Lloyd's K-Means Clustering Algorithm Is Frank-Wolfe in Disguise

Michael Pokojovy
Department of Mathematics & Statistics
and School of Data Science
Old Dominion University
Norfolk, VA 23529, USA
mpokojovy@odu.edu

J. Marcus Jobe
Information Systems &
Analytics Department
Miami University
Oxford, OH 45056, USA
jobejm@miamioh.edu

Simon Lacoste-Julien
Canada CIFAR AI Chair
Department of Computer Science and
Operations Research & Mila
Université de Montréal
Montréal (QC) Canada

Abstract

Lloyd's K-means algorithm, also known as naïve K-means, is a widely used *ad hoc* optimization heuristic, designed to minimize the sum of squared errors (SSE) across all K-partitions of a dataset via iterative cluster refinement. In this work, we establish a novel connection between Lloyd's algorithm and the Frank-Wolfe (FW) algorithm, a prominent first-order method for projection-free optimization. We demonstrate that Lloyd's algorithm is a special case of FW. Leveraging recent advances in FW methods for concave objectives, we derive a non-asymptotic $\mathcal{O}(1/t)$ convergence rate to a local minimum of the SSE objective. To account for empty clusters, an outcome possible under Lloyd's greedy assignment, we develop an FW variant for semismooth objectives while retaining the same convergence rate that is solely controlled by the initial SSE value. We illustrate our findings with a simulation study for spherical Gaussian mixtures and a real-world image segmentation dataset.

1 INTRODUCTION

Cluster analysis is a branch of unsupervised machine learning focused on partitioning unlabeled data into meaningful groups, or clusters, based on similarity or other relevant criteria. K-means clustering is one of the most popular and widely used clustering approaches. While K-means clustering is an NP-hard optimization problem (Arthur and Vassilvitskii, 2006), a variety of heuristic solution approaches, most notably Lloyd's algorithm (Lloyd, 1982), exist. While initialization and acceleration strategies can be quite nuanced and vary from algorithm to algorithm (Hamerly, 2010), the shared core of most implementations is the iterative reassignment strategy of Lloyd (1982). Regarded as one of the Top 10 algorithms in data mining (Wu et al., 2008), "the" K-means algorithm has been extensively studied in the literature and is nearly ubiquitous across various application fields. See Blömer et al. (2016); Ikotun et al. (2023); Steinley (2006) for thorough review and synthesis of notable results.

As a multi-faceted topic, K-means clustering has attracted significant attention from diverse communities including machine learning, data mining, computer science, statistics, optimization, etc. In this work, we are specifically interested in optimization aspects. A remarkable property of the minimization problem underlying K-means clustering is that it can be equivalently formulated using the language of combinatorial programming, mixed integer nonlinear programming, expectation maximization (EM), smooth and nonsmooth optimization, matrix factorization, etc. (Bagirov and Mohebi, 2015; Bauckhage, 2016; Bottou and Bengio, 1995). These paradigms offer unique insights into the nature of the problem and provide instruments of theoretical and practical importance. Continuing this promising pursuit, the present paper reveals another notable connection: *Lloyd's K-means algorithm is a special case of Frank-Wolfe (FW) algorithm.* In addition to mathematical elegance, this fact offers a unique opportunity to leverage the techniques of projection-free concave optimization in studying the convergence of Lloyd's K-means algorithm. This strategy has recently been successfully applied to analyzing Gaussian trimmed likelihood estimation (Pokojovy and Jobe, 2022) and the convex-concave procedure (Yurtsever and Sra, 2022) through the lens of FW algorithm.

Proceedings of the 29$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

Motivated by the semismooth concave nature of the K-means objective, we leveraged, adapted and improved some recent advancements in the field (Khamaru and Wainwright, 2019; Yurtsever and Sra, 2022) to address the challenges posed by Lloyd's K-means algorithm. The latter are specifically caused by empty clusters oftentimes resulting from Lloyd's greedy allocation that necessitates semismooth extension of otherwise smooth objective to an appropriate convex set.

Contributions. Main contributions of this work can be briefly summarized as follows:

- We show that the Lloyd's algorithm with greedy cluster assignment is a special case of the FW algorithm with unit step-size for a semismooth concave objective over a polyhedral set. With empty clusters excluded from the feasible set, the objective becomes smooth. This observation allows us to leverage recent convergence results for concave FW to directly establish a non-asymptotic $\mathcal{O}(1/t)$ convergence rate (Yurtsever and Sra, 2022).
- To accommodate for the practically relevant possibility of empty clusters, we develop an FW algorithm for general (semismooth) concave objectives with a new FW gap based on Clarke subdifferential. The same non-asymptotic $\mathcal{O}(1/t)$ convergence rate is recovered that solely depends on the initial global suboptimality (uniformly in sample size n, space dimension d and number of clusters K) and does not involve coreset considerations.

In addition to self-contained proofs, a simulation study was performed to corroborate these results.

Relevance. Due to its importance, the K-means problem has been extensively studied both in terms of algorithmic complexity and convergence speed of existing computational heuristics (Blömer et al., 2016) and remains an active research field. The ability to view Lloyd's K-means algorithm through the lens of semismooth FW algorithm gives a new simple, yet powerful tool that provides both convergence guarantees and a practical error control mechanism. Outside of core domain, the results are expected to have implications for more general forms of robust cluster analysis aimed to account for the presence of outliers and other model violations (Dorabiala et al., 2022; García-Escudero et al., 2008). With limited amount of convergence results available to date, primarily due to unique combinatorial challenges associated with computing robust estimators (Bernholt and Fischer, 2004), our new approach promises to have even more significance. Lastly, our work is potentially relevant for any situation, where greedy variants of the linear minimization oracle (LMO) can destroy smoothness.

Setup. The K-means SSE objective reads as

$$f(\boldsymbol{w}) = \sum_{k=1}^{K} \sum_{i=1}^{n} w_{ik} \|\boldsymbol{x}_i - \boldsymbol{x}_k^{\boldsymbol{w}}\|^2 \tag{1}$$

with $\boldsymbol{x}_k^{\boldsymbol{w}} := \frac{1}{n_k^{\boldsymbol{w}}} \sum_{i=1}^{n} w_{ik} \boldsymbol{x}_i$ and $n_k^{\boldsymbol{w}} := \sum_{i=1}^{n} w_{ik}$ using the usual 'one-hot' encoding $\boldsymbol{w} \in \{0,1\}^{n \times K}$ such that $w_{ik} = 1$ if the i-th data point $\boldsymbol{x}_i \in \mathbb{R}^d$ is in the k-th cluster. The goal is to minimize f over all partitions $\boldsymbol{w} \in \mathcal{M}_0^{\text{adm}}$ with non-overlapping non-empty clusters. Since f is concave (cf. Lemma 6), the latter is equivalent to solving

$$\min_{\boldsymbol{w} \in \mathcal{M}^{\text{adm}}} f(\boldsymbol{w}) \tag{2}$$

over the convex set $\mathcal{M}^{\text{adm}} := \text{conv}(\mathcal{M}_0^{\text{adm}})$ (Horst, 1984). Since Lloyd's K-means cluster assignment corresponds to FW stepping with a unit step-size in the direction provided by a greedy linear minimization oracle (LMO) over a larger convex set $\mathcal{M}$

$$\min_{\boldsymbol{w} \in \mathcal{M}} f(\boldsymbol{w}), \tag{3}$$

in which the objective f is still concave, but only semismooth, the latter configuration is adopted and analyzed in this paper.

2 RELATED WORK

Frank-Wolfe Algorithm. Originally proposed by Frank and Wolfe (1956), the FW algorithm, also known as conditional gradient descent, has recently experienced a major resurgence in interest in the machine learning community due to wide applicability and suitability for large-scale machine learning problems, adversarial learning, sparse estimation in multiple linear regression, support vector machine training, matrix completion, computational geometry, etc. See (Jaggi, 2013; Pokutta, 2023; Bomze et al., 2020) for applications and introduction to projection-free optimization with FW.

The FW algorithm (cf. Algorithm 1) aims to solve optimization problem

$$\min_{\boldsymbol{x} \in \mathcal{M}} f(\boldsymbol{x}) \tag{4}$$

for a convex compact $\emptyset \neq \mathcal{M} \subset \mathbb{R}^d$. Unlike projection-based methods (Nesterov, 2018), orthogonal projectors are replaced with linear minimization oracle (LMO)

$$\text{LMO}_{\mathcal{M}}(\boldsymbol{r}) := \arg\min_{\boldsymbol{s} \in \mathcal{M}} \langle \boldsymbol{s}, \boldsymbol{r} \rangle. \tag{5}$$

Classical stepping mechanisms include the original fixed step-size scheme of (Frank and Wolfe, 1956), line search or adaptive stepping subject to a curvature constant bound (Lacoste-Julien, 2016). See (Bomze et al., 2019, 2020; Locatello et al., 2017; Négiar et al., 2020) for alternative strategies. One of the key quantities in the "smooth" FW algorithm is the FW gap $g_t \coloneqq g(\boldsymbol{x}^{(t)})$ with

$$g(\boldsymbol{x}) \equiv \max_{\boldsymbol{s}\in\mathcal{M}} \big\langle \boldsymbol{s}-\boldsymbol{x}, -\nabla f(\boldsymbol{x})\big\rangle \quad \text{at the } t\text{-th iterate.} \tag{6}$$

The FW gap $g(\boldsymbol{x})$ is a special case of the duality gap and, thus, can be used to gauge stationarity.

Various assumptions on the feasible set $\mathcal{M}$ and the objective f are encountered in the literature. While $\mathcal{M}$ is usually assumed as a generic non-empty convex compact set—although uniform convex sets (Garber and Hazan, 2015) or polytopes (GuéLat and Marcotte, 1986) are sometimes specifically considered—the requirements on f may vary depending on the stepping strategy, desired convergence rate and other factors. Most of the existing research focuses on convex or strongly convex smooth objectives (Garber and Hazan, 2015, 2016; Lacoste-Julien and Jaggi, 2015; Pedregosa et al., 2020). Some results vary depending on whether the optimum $\boldsymbol{x}^*$ is a boundary or an interior point.

While there has been growing interest in applying the Frank-Wolfe method to non-convex optimization, primarily due to new applications in multiple sequence alignment (Alayrac et al., 2016, Appendix B), multi-object tracking (Chari et al., 2015, Section 5.1), robust estimation of Gaussian models (Pokojovy and Jobe, 2022), fewer results exist compared to the convex case. While earlier works (e.g., Dunn (1979)) assumed invexity, general non-convex objectives with finite curvature have recently been studied (Lacoste-Julien, 2016; Pedregosa et al., 2020; Reddi et al., 2016) leading to $\min_{0\le\tau\le t} g_\tau = \mathcal{O}(1/\sqrt{t})$ (or slower rates for less smooth functions (de Oliveira, 2023)) under various step control strategies. The rate is generally inferior to the usual convex rate of $\mathcal{O}(1/t)$. See (Bomze et al., 2020, Table 2), (Garber and Hazan, 2015, Table 1) and (Pedregosa et al., 2020, Table 1) for summary. Some other works specifically focus on concave objectives (Clarkson, 2010; Mangasarian, 1996; Rinaldi et al., 2009) with a recently established $\mathcal{O}(1/t)$ convergence rate (Yurtsever and Sra, 2022) under the greedy unit step-size choice. Another important research avenue is FW algorithm for nonsmooth functions (Krishnan et al., 2015; Khamaru and Wainwright, 2019; Ravi et al., 2019), including for concave objectives (White, 1993) like the one we investigate in Section 3. Our work closes a gap in the existing literature by developing an FW algorithm for concave semismooth objectives while relying on Clarke subdifferential in lieu of the less convenient Goldstein's subgradient and preserving recent $\mathcal{O}(1/t)$ convergence results in the smooth case.

K-means Clustering as Continuous Optimization. K-means clustering has been extensively studied from the viewpoint of continuous optimization (Bagirov and Mohebi, 2015; Bottou and Bengio, 1995; Selim and Ismail, 1984). Unrelated to Lloyd's algorithm, Bauckhage (2016) proposed to use FW with diminishing step as an optimization heuristic for K-means. Another major connection follows from the theory of ε-coresets (Clarkson, 2010; Har-Peled and Kushal, 2007; Ravi et al., 2019). In addition to being quite involved, estimates derived for coresets, although typically bounded in the sample size n and dimension d, tend to explode as $\varepsilon \to 0$. The novelty of our work is the ability to derive an $\mathcal{O}(1/t)$ convergence rate that solely depends on the initial suboptimality and holds uniformly in sample size n, dimension d and number of clusters K. Leveraging the properties of FW algorithm in general convex geometries, no polytope structure or combinatorial arguments are involved which makes our approach robust to modifications and extensions, which recently proved important in other areas (Pokojovy and Jobe, 2022; Yurtsever and Sra, 2022).

3 FW FOR SEMISMOOTH CONCAVE OBJECTIVES

Consider the optimization problem in Equation (4). Assume that $\mathcal{M} \subset \mathbb{R}^d$ is a non-empty compact convex set and $f\colon \mathcal{M} \to \mathbb{R}$ is a concave continuous function—not necessarily continuously differentiable in $\mathcal{M}$. The subdifferential ∂f of f can be defined as the negative of the subdifferential of the convex function $-f$. Moreover, this definition agrees with other three standard concepts of limiting, Frechét and Clarke subdifferentials (Rockafellar and Wets, 1998, Chapters 8-9). With a slight notation abuse, for simplicitly, we will denote all three subdifferentials by ∂f. Further, the set $\partial f(\boldsymbol{x}) \neq \emptyset$ is compact and convex for any $\boldsymbol{x} \in \mathcal{M}$ and, if f is differentiable at $\boldsymbol{x}$, $\partial f(\boldsymbol{x}) = \{\nabla f(\boldsymbol{x})\}$ is a singleton.

FW algorithm for semismooth functions has recently been proposed and investigated by Khamaru and Wainwright (2019). Under the umbrella of semismooth functions, various not always equivalent alternative definitions exist (Rockafellar and Wets, 1998) with the shared goal of generalizing the classical subdifferential of a convex function to more general classes. Introducing a curvature constant akin to Lacoste-Julien (2016) but using Frechét subdifferential, Rockafellar and Wets (1998) showed an $\mathcal{O}(1/\sqrt{t})$ convergence of FW with variable step-sizes without rigorously explaining the

stationarity concept adopted, which is provided in this paper. Additionally, assuming the objective is concave, we show an improved $\mathcal{O}(1/t)$ convergence rate by adapting the proof of Yurtsever and Sra (2022) presented for smooth objectives. Algorithm 1 protocols our semismooth FW with unit step-size. Note that for smooth objectives, it automatically reduces to the smooth version since $\partial f(\boldsymbol{x}) = \{\nabla f(\boldsymbol{x})\}$.

Algorithm 1 (Semismooth FW for concave functions)**.**

Let $\boldsymbol{x}^{(0)} \in \mathcal{M}$.
for $t = 0, \dots, T-1$ ***do***
  Choose arbitrary $\boldsymbol{r}^{(t)} \in \partial f(\boldsymbol{x}^{(t)})$.
  Compute $\boldsymbol{s}^{(t)} := \mathrm{LMO}_{\mathcal{M}}\left(\boldsymbol{r}^{(t)}\right)$.
  Define the FW update direction $\boldsymbol{d}_t := \boldsymbol{s}^{(t)} - \boldsymbol{x}^{(t)}$.
  Compute the FW gap $g_t := \left\langle \boldsymbol{d}_t, -\boldsymbol{r}^{(t)} \right\rangle$.
  if $g_t \leq \epsilon$ ***then***
    return $\boldsymbol{x}^{(t)}$
  end if
  Select step-size $\gamma_t = 1$.
  Update $\boldsymbol{x}^{(t+1)} := \boldsymbol{x}^{(t)} + \gamma_t \boldsymbol{d}_t \equiv \boldsymbol{s}^{(t)}$.
end for
return $\boldsymbol{x}^{(T)}$.

Before discussing non-asymptotic convergence of Algorithm 1, we briefly introduce a suitable generalization of the FW gap to be used as a stationarity measure. Instead of relying on (δ, ϵ)-Goldstein stationarity (Liu et al., 2024; Ravi et al., 2019), we employ the concept of Clarke stationarity. Recall that a point $\boldsymbol{x} \in \mathcal{M}$ is referred to as Clarke-stationary (de Oliveira, 2023) if

$$0 = \max_{\boldsymbol{y} \in \mathcal{M}} \langle \boldsymbol{y} - \boldsymbol{x}, -\boldsymbol{r} \rangle = 0 \tag{7}$$

for at least one $\boldsymbol{r} \in \partial f(\boldsymbol{x})$.

Similar to Liu et al. (2024), we introduce the Clarke version of FW gap

$$g_{\mathrm{C}}(\boldsymbol{x}) := \min_{\boldsymbol{r} \in \partial f(\boldsymbol{x})} \max_{\boldsymbol{y} \in \mathcal{M}} \langle \boldsymbol{y} - \boldsymbol{x}, -\boldsymbol{r} \rangle. \tag{8}$$

By compactness, the latter expression is well-defined and finite. Further, for any $\boldsymbol{r} \in \partial f(\boldsymbol{x})$,

$$\max_{\boldsymbol{y} \in \mathcal{M}} \langle \boldsymbol{y} - \boldsymbol{x}, -\boldsymbol{r} \rangle \geq \langle \boldsymbol{y} - \boldsymbol{x}, -\boldsymbol{r} \rangle|_{\boldsymbol{y} = \boldsymbol{x}} = 0.$$

Thus, we trivially have $g_{\mathrm{C}}(\boldsymbol{x}) \geq 0$. By Equation (7), $g_{\mathrm{C}}(\boldsymbol{x}) = 0$ if and only if $\boldsymbol{x}$ is Clarke stationary.

As for smooth functions (Yurtsever and Sra, 2022), Lemma 2 shows the usual line search in FW is equivalent with the greedy choice of a unit step-size. This offers a major computational advantage since no additional function evaluations are required to solve the directional minimization problem.

Lemma 2. *The unit step-size* $\gamma_t = 1$ *solves the line search problem*

$$\min_{\gamma \in [0,1]} f\left(\boldsymbol{x}^{(t)} + \gamma \boldsymbol{d}_t\right).$$

Proof. Since f is a concave function, so is $\varphi_t \colon [0,1] \to \mathbb{R}$, $\varphi_t(\gamma) := f\left(\boldsymbol{x}^{(t)} + \gamma \boldsymbol{d}_t\right)$ at each step t of Algorithm 1. Therefore, without necessarily being smooth, φ_t must attain its global (possibly non-strict) minima at the boundary $\{0, 1\}$ (Horst, 1984). Arguing as in Equation (11) below and recalling that $g_t \geq 0$, we have $\varphi_t(1) \leq \varphi_t(0)$, whence φ_t has a minimum at $\gamma = 1$. □

Theorem 3 (Convergence of semismooth FW on concave objectives)**.** *Consider running the FW Algorithm 1 with unit step-size for the optimization problem in Equation (4). Then the minimal FW gap* $\tilde{g}_t := \min_{0 \leq \tau \leq t} g_\tau$ *encountered by the iterates after* t *iterations satisfies:*

$$\tilde{g}_t \leq \frac{h_0}{t+1} \quad \text{for } 0 \leq t \leq T-1 \tag{9}$$

where $h_0 := f(\boldsymbol{x}^{(0)}) - \min_{\boldsymbol{x} \in \mathcal{M}} f(\boldsymbol{x})$ *is the initial global suboptimality.*

The proof follows the streamlines of Yurtsever and Sra (2022) in the smooth case (see also Lacoste-Julien (2016)). At each step of the algorithm, the objective is decreased by (at least) the FW gap g_t. As cumulative reduction is bounded by the initial global suboptimality over $\mathcal{M}$ (or, equivalently, $\partial\mathcal{M}$), the FW g_t must eventually become small. Compared to the general non-convex case, the concavity of f not only furnishes an improved rate of $\mathcal{O}(1/t)$ in lieu of $\mathcal{O}(1/\sqrt{t})$ but produces a bound that solely depends on the initial suboptimality.

Proof. For arbitrary $\gamma \in [0,1]$, compute the point $\boldsymbol{x}_\gamma := \boldsymbol{x}^{(t)} + \gamma \boldsymbol{d}^{(t)}$ by moving with step-size γ in direction $\boldsymbol{s}^{(t)}$, where $\boldsymbol{d}_t := \boldsymbol{s}^{(t)} - \boldsymbol{x}^{(t)}$ is the FW direction as defined by Algorithm 1. By concavity,

$$f(\boldsymbol{x}_\gamma) \leq f(\boldsymbol{x}^{(t)}) + \gamma \langle \boldsymbol{d}_t, \boldsymbol{r} \rangle \text{ for all } \boldsymbol{r} \in \partial f(\boldsymbol{x}^{(t)}). \tag{10}$$

Letting $\gamma = 1$ and $\boldsymbol{r} = \boldsymbol{r}^{(t)}$, we get

$$f(\boldsymbol{s}^{(t)}) \leq f(\boldsymbol{x}^{(t)}) - \langle \boldsymbol{d}_t, -\boldsymbol{r}^{(t)} \rangle = f(\boldsymbol{x}^{(t)}) - g_t \tag{11}$$

for $t = 0, \dots, T-1$. By induction,

$$f\left(\boldsymbol{x}^{(t+1)}\right) \leq f\left(\boldsymbol{x}^{(0)}\right) - \sum_{\tau=0}^{t} g_\tau \tag{12}$$

for $t = 0, \dots, T-1$ and, therefore,

$$\begin{aligned} f\left(\boldsymbol{x}^{(t+1)}\right) &\leq f\left(\boldsymbol{x}^{(0)}\right) - (t+1) \min_{0 \leq \tau \leq t} g_\tau \\ &= f\left(\boldsymbol{x}^{(0)}\right) - (t+1)\tilde{g}_t \text{ for } t = 0, \dots, T \end{aligned} \tag{13}$$

with $\tilde{g}_t := \min_{0 \leq \tau \leq t} g_\tau$. Estimating for $t = 0, \dots, T-1$

$$f(\boldsymbol{x}^{(0)}) - f(\boldsymbol{x}^{(t+1)}) \leq f(\boldsymbol{x}^{(0)}) - \min_{\boldsymbol{x} \in \mathcal{M}} f(\boldsymbol{x}) \equiv h_0$$

and solving Equation (13) for $\tilde{g}_t$, the claim follows. □

Remark 4. *In case of ambiguity, the LMO in FW Algorithm 1, line 4 can be amended to always return an extreme point of $\mathcal{M}$ (Horst, 1984). This property is trivially inherited by the iterates $\boldsymbol{x}^{(t)}$ for $t = 1, \dots, T$. Also, if $\mathcal{M}$ is a convex polytope, i.e., $\mathcal{M} = \operatorname{conv}(\mathcal{M}_0)$ for some finite $\mathcal{M}_0 \neq \emptyset$, one can similarly argue the FW Algorithm 1 must terminate in $|\mathcal{M}_0|$ steps or less.*

4 LLOYD'S K-MEANS AS FW FOR CONCAVE OBJECTIVE

We apply the general results of Section 3 to the K-means clustering. Starting with the case of non-empty clusters, Lemma 6 below characterizes the SSE objective as a smooth concave function on a compact set, directly producing an $\mathcal{O}(1/t)$ convergence in the FW gap. In case of empty cluster(s), the argumentation is somewhat more involved. For any cluster turning empty at some iteration t, an "ephemerous" center can selected using the previous center, performing random sampling from the dataset or using another strategy. We show that any such selection strategy corresponds to selecting an element of the Clarke subdifferential of the extended semismooth concave objective, which, in turn, yields the same $\mathcal{O}(1/t)$ convergence rate.

Consider a dataset $\mathcal{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\} \subset \mathbb{R}^d$. We seek to minimize the sum of squares errors (SSE), also known as within-cluster sum of squares (WCSS),

$$\operatorname{SSE}(\mathcal{C}) \equiv \sum_{k:C_k \neq \emptyset} \sum_{\boldsymbol{x} \in C_k} \|\boldsymbol{x} - \bar{\boldsymbol{x}}_k\|^2 \tag{14}$$

with $\bar{\boldsymbol{x}}_k = \frac{1}{|C_k|} \sum_{\boldsymbol{x} \in C_k} \boldsymbol{x}$ over all K-partitions $\mathcal{C} = (C_1, C_2, \dots, C_K)$ of $\mathcal{D}$ of size $K \in \mathbb{N}$.

Algorithm 5 (Lloyd's K-means algorithm)**.**

Let the seeds $\boldsymbol{\mu}_1^{(-1)}, \dots, \boldsymbol{\mu}_K^{(-1)} \in \mathbb{R}^d$ be given.
Initialize:
$C_k^{(0)} := \{\boldsymbol{x}_i \,|\, \forall k' : \|\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(-1)}\| \leq \|\boldsymbol{x}_i - \boldsymbol{\mu}_{k'}^{(-1)}\|\}$,
$\boldsymbol{\mu}_k^{(0)} := \begin{cases} \frac{1}{|C_k^{(0)}|} \sum_{\boldsymbol{x}_i \in C_k^{(0)}} \boldsymbol{x}_i, & \text{if } C_k^{(0)} \neq \emptyset, \\ \boldsymbol{\mu}_k^{(0)} := \boldsymbol{\mu}_k^{(-1)}, & \text{otherwise.} \end{cases}$
for $t = 0, \dots, T-1$ ***do***
  for $k = 1, \dots, K$ ***do***
    Let
    $C_k^{(t+1)} := \{\boldsymbol{x}_i \,|\, \forall k' : \|\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(t)}\| \leq \|\boldsymbol{x}_i - \boldsymbol{\mu}_{k'}^{(t)}\|\}$.
    Compute
    $\boldsymbol{\mu}_k^{(t+1)} := \begin{cases} \frac{1}{|C_k^{(t+1)}|} \sum_{\boldsymbol{x}_i \in C_k^{(t+1)}} \boldsymbol{x}_i, & \text{if } C_k^{(t+1)} \neq \emptyset, \\ \boldsymbol{\mu}_k^{(t)}, & \text{otherwise.} \end{cases}$
    Let $\Delta\operatorname{SSE}_t := \operatorname{SSE}(\mathcal{C}^{(t)}) - \operatorname{SSE}(\mathcal{C}^{(t+1)})$.
    if $\Delta\operatorname{SSE}_t \leq \epsilon$ ***then***
      return $C_1^{(t)}, \dots, C_K^{(t)}$ *(and* $\boldsymbol{\mu}_1^{(t)}, \dots, \boldsymbol{\mu}_K^{(t)}$*).*
    end if
  end for
end for
return $C_1^{(T)}, \dots, C_K^{(T)}$ *(and* $\boldsymbol{\mu}_1^{(T)}, \dots, \boldsymbol{\mu}_K^{(T)}$*).*

Algorithm 5 (cf. (MacKay, 2003, Algorithm 20.3)) protocols one of the common formulations of the Lloyd's K-means algorithm – with some trivial adjustments to make the indexing consistent with that of FW algorithm. Starting with some initial partition, the algorithm proceeds with forming K clusters by assigning points $\boldsymbol{x}_i$ to the closest center (ties being broken arbitrarily). For all non-empty clusters, respective means are updated before proceeding to the next iteration. As initial partition, Algorithm 5 implements the usual Lloyd's initialization, but our convergence results to any other initialization strategy, e.g., based on optimized seeding strategies (Arthur and Vassilvitskii, 2007). For any given run, the usual $\mathcal{O}(ndKT)$ complexity and $\mathcal{O}(nd + Kd)$ storage hold.

Our goal is to analyze Algorithm 5 through the prism of FW. To this end, every partition $\mathcal{C}$ of the dataset $\mathcal{D}$ is uniquely encoded by a vector $\boldsymbol{w} \in \mathcal{M}_0$ such that $w_{ik} = 1$ if and only if $\boldsymbol{x}_i \in C_k$, where

$$\mathcal{M}_0 = \Big\{\boldsymbol{w} \in \{0,1\}^{n \times K} \,\Big|\, \sum_{k=1}^{K} w_{ik} = 1, i = 1, \dots, n\Big\}. \tag{15}$$

The SSE objective can be continuously extended to $\mathcal{M} = \operatorname{conv}(\mathcal{M}_0)$ via

$$f(\boldsymbol{w}) = \sum_{k=1}^{K} \sum_{i=1}^{n} w_{ik} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}\|^2 \tag{16}$$

with $\boldsymbol{x}_k^{\boldsymbol{w}} := \frac{1}{n_k^{\boldsymbol{w}}} \sum_{i=1}^{n} w_{ik} \boldsymbol{x}_i$ and $n_k^{\boldsymbol{w}} := \sum_{i=1}^{n} w_{ik}$, where $\sum_{i=1}^{n} w_{ik} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}\|^2 := 0$ is set for each k with $n_k^{\boldsymbol{w}} = 0$. Lemma 6 gives the gradient and the Hessian of f (proof in Supplemental Section A).

Lemma 6. *For any $\boldsymbol{w} \in \mathcal{M}$ with $n_k^{\boldsymbol{w}} > 0$ for all k, ∇f and $\nabla^2 f$ read as*

$$\partial_{w_{ik}} f(\boldsymbol{w}) = \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}\|^2 \quad \text{and}$$
$$\partial_{w_{ik}} \partial_{w_{jl}} f(\boldsymbol{w}) = -\frac{2}{n_k^{\boldsymbol{w}}} \cdot \mathbb{1}_{\{k=l\}} \cdot (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}) \cdot (\boldsymbol{x}_j - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}).$$

Moreover, the Hessian $\nabla^2 f$ is negative semidefinite.

Hence, f is concave in the interior of $\mathcal{M}$ and, due to Bauer's minimum principle (Kružík, 2000), attains a global minimum at one of the extreme points $\mathcal{M}_0$. This also applies to any local minimum that either must be an extreme point itself or be "tied" with one. Unfortunately, ∇f can not be continuously extended to $\mathcal{M}$ so that the smooth FW algorithm cannot be applied to f over $\mathcal{M}$.

Non-Empty Clusters. For illustration purposes, before analyzing the general case, we first consider the "admissible" version of $\mathcal{M}_0$ where all clusters are assumed non-empty, namely:

$$\mathcal{M}_0^{\text{adm}} = \Big\{ \boldsymbol{w} \in \{0,1\}^{n\times K} \mid \sum_{k=1}^{K} w_{ik} = 1, i = 1,\dots,n, \quad \sum_{i=1}^{n} w_{ik} \geq 1, k = 1,\dots,K \Big\}, \tag{17}$$

and define the polytope $\mathcal{M}^{\text{adm}} := \operatorname{conv}\big(\mathcal{M}_0^{\text{adm}}\big)$. By Lemma 6, f is smooth in $\mathcal{M}^{\text{adm}} \subset \mathbb{R}^{n\times K}$.

Remark 7. *Though irrelevant due to concavity, $\nabla^2 f(\boldsymbol{w})$ is bounded for $\boldsymbol{w} \in \mathcal{M}^{\text{adm}}$ (see Supplement):*

$$\|\nabla^2 f(\boldsymbol{w})\| \leq 2R^2 \quad \text{with} \quad R = \max_{i,j=1,\dots,n} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|.$$

Therefore, f has a bounded curvature constant

$$C_f = 4n\big(\max_{i,j=1,\dots,n} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\big)$$

over $\mathcal{M}^{\text{adm}}$. Thus, the general non-convex approach of Lacoste-Julien (2016) is applicable but furnishes an inferior $\mathcal{O}(1/\sqrt{t})$ rate compared to the concave rate $\mathcal{O}(1/t)$ of Yurtsever and Sra (2022).

We endow $\mathbb{R}^{n\times K}$ with the usual Frobenius scalar product $\langle \boldsymbol{w}, \tilde{\boldsymbol{w}} \rangle_{\mathcal{F}} = \operatorname{tr}(\boldsymbol{w}'\tilde{\boldsymbol{w}})$. With the space $\mathcal{F}$ being isometrically isomorphic to $\mathbb{R}^{nK}$, (Yurtsever and Sra, 2022, Lemma 2.1) is applicable. Interestingly, since FW is affine-invariant (Jaggi, 2013), the choice of the scalar product does not affect the results.

The FW update in Algorithm 1 with the unit step-size $\gamma_t = 1$ reads as

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{s}^{(t)} := \operatorname*{arg\,min}_{\boldsymbol{s}\in\mathcal{M}^{\text{adm}}} \langle \boldsymbol{s}, \nabla f(\boldsymbol{w}^{(t)}) \rangle_{\mathcal{F}}. \tag{18}$$

Using ∇f from Lemma 6, the latter linear optimization problem can be explicitly expressed as

$$\min_{\boldsymbol{s}\in\mathcal{M}_0^{\text{adm}}} \sum_{k=1}^{K}\sum_{i=1}^{n} s_{ik} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}^{(t)}}\|^2. \tag{19}$$

The greedy solution $\tilde{\boldsymbol{s}}^{(t)} \in \mathcal{M}_0$ is given by

$$\tilde{s}_{ik}^{(t)} = \begin{cases} 1, & \bar{\boldsymbol{x}}_k = \operatorname*{arg\,min}_{k'=1,\dots,K} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_{k'}^{\boldsymbol{w}^{(t)}}\|^2 \\ 0, & \text{otherwise} \end{cases} = \begin{cases} 1, & \bar{\boldsymbol{x}}_k = \operatorname*{arg\,min}_{k'=1,\dots,K} \|\boldsymbol{x}_i - \boldsymbol{\mu}_{k'}^{(t)}\|^2 \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

where we observed

$$\boldsymbol{\mu}_k^{(t)} = \bar{\boldsymbol{x}}_k^{\boldsymbol{w}^{(t)}} \quad \text{for} \quad \boldsymbol{w}^{(t)} \in \mathcal{M}^{\text{adm}}.$$

Thus, the assignment in Equation (20) equivalent with line 8 of Algorithm 5.

For all $\boldsymbol{s} \in \mathcal{M}$, we can trivially estimate

$$\sum_{i=1}^{n}\sum_{k=1}^{K} s_{ik}\|\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(t)}\|^2 \geq \sum_{i=1}^{n}\sum_{k=1}^{K} s_{ik} \min_{k'} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(t)}\|^2 = \sum_{i=1}^{n} \min_{k'} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(t)}\|^2 = \sum_{i=1}^{n}\sum_{k=1}^{K} \tilde{s}_{ik}^{(t)} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(t)}\|^2.$$

Thus, $\tilde{\boldsymbol{s}}^{(t)}$ solves Equation (19) over the superset $\mathcal{M}$. However, as long as $\tilde{\boldsymbol{s}}^{(t)} \in \mathcal{M}^{\text{adm}}$, i.e., no cluster is empty, it also solves the original problem (19). Invoking (Yurtsever and Sra, 2022, Lemma 2.1), the FW gap g_t is equal to the decrease in the objective f at step $0 \leq t \leq T-1$:

$$g_t = \big\langle \boldsymbol{d}_t, -\nabla f(\boldsymbol{w}^{(t)}) \big\rangle = \big\langle \boldsymbol{w}^{(t+1)} - \boldsymbol{w}^{(t)}, -\nabla f(\boldsymbol{x}^{(t)}) \big\rangle = \sum_{k=1}^{K}\sum_{i=1}^{n} w_{ik}^{(t)} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}^{(t)}}\|^2 - \sum_{k=1}^{K}\sum_{i=1}^{n} w_{ik}^{(t+1)} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}^{(t)}}\|^2 = f\big(\boldsymbol{x}^{(t)}\big) - f\big(\boldsymbol{x}^{(t+1)}\big).$$

With the preparations above, the next Theorem is a direct consequence of (Yurtsever and Sra, 2022, Lemma 2.1)—or our more general Theorem 3—applied to the concave SSE objective of the K-means algorithm.

Theorem 8. *Let $\boldsymbol{w}^{(t)}$, $0 \leq t \leq T$, be produced by Lloyd's K-means algorithm. Assuming $\boldsymbol{w}^{(t)} \in \mathcal{M}_0^{\text{adm}}$ for all t, the membership vector $\boldsymbol{w}^{(t)}$ coincides with the the t-th iterate of the FW Algorithm 1 applied to minimizing f over $\mathcal{M}^{\text{adm}}$ so that the running minimum FW gap satisfies*

$$\tilde{g}_t = \min_{0\leq\tau\leq t} g_\tau = \min_{0\leq\tau\leq t} \Delta\text{SSE}_\tau \leq \frac{h_0}{t+1}$$

for $t = 0,\dots,T-1$, where $h_0 = f(\boldsymbol{w}^{(0)}) - \min_{\boldsymbol{w}\in\mathcal{M}} f(\boldsymbol{w})$ and $\Delta\text{SSE}_t = f\big(\boldsymbol{w}^{(t)}\big) - f\big(\boldsymbol{w}^{(t+1)}\big)$.

It should be emphasized that the (unit) constant in Theorem 8 is independent of the sample size n, the dimension p or initial clusters. Unlike proofs based on ε-coreset, no dependence on ε occurs. The initial suboptimality h_0 itself scales linearly in n and p. Invoking the standard SSE identity (Johnson and Wichern, 2007, p. 693) and $f(\boldsymbol{w}) \geq 0$, the initial gap $h_0 := f(\boldsymbol{w}^{(0)}) - \min_{\boldsymbol{w}\in\mathcal{M}} f(\boldsymbol{w})$ can easily be estimated via

$$h_0 \leq (n-1)\operatorname{tr}(\boldsymbol{S}_n) \equiv \sum_{i=1}^{n} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}\|^2 \tag{21}$$

where $\bar{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i$ and $\boldsymbol{S}_n := \frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})'$ are the usual sample mean vector and (unbiased) sample covariance matrix of the entire dataset $\mathcal{D}$.

Remark 9. *If the data $\boldsymbol{x}_i$'s are independently sampled from a squared integrable distribution on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with covariance matrix $\boldsymbol{\Sigma}$, the strong law of large numbers implies*

$$h_0 = (n-1)\,\mathrm{tr}\left(\boldsymbol{\Sigma}\right) + o_{\mathbb{P}}(n) \quad \mathbb{P}\text{-a.s. as } n \to \infty$$

uniformly over arbitrary initial cluster choice.

General Case. If Lloyd's greedy assignment fails to produce admissible cluster membership, the linear programming problem in Equation (18) furnishes an alternative form of Lloyd's update that guarantees the cluster allocation stays "admissible" at any time. Since no closed-form closed solution exists, Thorup's version of the Hungarian algorithm (Thorup, 2004) can be used to compute the solution in an $\mathcal{O}(nK^2 + K^2 \log\log(K))$-time. This approach is generally disfavored in most communities. In addition to computational considerations, the rationale is best illustrated by non-spherical Gaussian mixture models. Indeed, even clusters with $d+1$ or more points in general position are not guaranteed to produce well-conditioned covariances (García-Escudero et al., 2008). Thus, constraints on cluster sizes, including non-empty clusters, are often viewed as ineffective.

Instead of amending the Lloyd's K-means algorithm, we will pursue a different approach here by analyzing it through the framework of semismooth FW Algorithm 1. In addition to being continuous on $\mathcal{M}$, f is concave and continuously differentiable for all $\boldsymbol{w} \in \mathcal{M}$ with $n_k^{\boldsymbol{w}} > 0$ for all $k = 1, \ldots, K$ and, therefore, semismooth on $\mathcal{M}$.

For any $\boldsymbol{w} \in \mathcal{M}$ with $n_k^{\boldsymbol{w}} = 0$ for at least one (but at most $K-1$) k, let $\boldsymbol{c}_k \in \mathcal{M}$ denote a prescribed "ephemerous" center of the k-th cluster. Consider any sequence $(\boldsymbol{w}_j)_j \subset \mathrm{int}(\mathcal{M})$ such that

$$\lim_{j\to\infty} \boldsymbol{w}_j = \boldsymbol{w} \quad \text{and} \quad \lim_{j\to\infty} \bar{\boldsymbol{x}}_k^{\boldsymbol{w}_j} = \boldsymbol{c}_k \tag{22}$$

for every k with $n_k^{\boldsymbol{w}} = 0$. Such selection is easily possible since we can write

$$\boldsymbol{c}_k = \frac{1}{\sum_{i=1}^{n} \alpha_{ik}} \sum_{i=1}^{n} \alpha_{ik} \boldsymbol{x}_i$$

for some $\alpha_{ik} \in [0,1]$ and rescale α_{ik}'s to enforce the condition $\sum_{i=1}^{n} \alpha_{ik} = o(1)$ while adjusting other columns by $o(1)$ to maintain the remaining constraints in $\mathcal{M}$. By Lemma 6,

$$\lim_{j\to\infty} \partial_{w_{ik}} f(\boldsymbol{w}^j) = \begin{cases} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}\|^2, & n_k^{\boldsymbol{w}} > 0, \\ \|\boldsymbol{x}_i - \boldsymbol{c}_k\|^2, & \text{otherwise.} \end{cases}$$

As a directional derivative, the latter is an element of $\partial f(\boldsymbol{w})$. Importantly, this is exactly the subgradient $\boldsymbol{r}^{(t)} \in \partial f(\boldsymbol{w}^{(t)})$ used by Lloyd's greedy LMO over $\mathcal{M}$ as part of Algorithm 5 if we let $\boldsymbol{c}_k = \boldsymbol{\mu}_k^{(t)}$ at the t-th step. Moreover, we also have $\boldsymbol{\mu}_k^{(t)} = \bar{\boldsymbol{x}}_k^{\boldsymbol{w}^{(t)}}$ if $n_k^{\boldsymbol{w}^{(t)}} > 0$.

Thus, we arrive at a nonsmooth generalization of Theorem 8, for which we can now drop the assumption of $\boldsymbol{w}^{(t)} \in \mathcal{M}^{\mathrm{adm}}$ for $t = 0, \ldots, T$ while retaining the same convergence estimate.

Practical Implications. Our theoretical developments put forward a new *bona fide* error measure

$$g_t \equiv \Delta\mathrm{SSE}_t \tag{23}$$

that can be used in stopping decisions when running the K-means algorithm. Presently, the root of the sum of squared distances between new and old centroids

$$\epsilon_t := \Big(\sum_{k=1}^{K} \big\|\boldsymbol{\mu}_k^{(t+1)} - \boldsymbol{\mu}_k^{(t)}\big\|^2\Big)^{1/2} \tag{24}$$

is typically employed as an error measure.

Unlike our FW gap-based measure (Lacoste-Julien, 2016), the error measure in Equation (24) is not affine invariant and has no known convergence rate. Further, the choice of the tolerance in the stopping rule is problematic since the right scale is difficult to guess, which may lead to premature stopping or longer than necessary runs. In contrast, two natural choices of the tolerance threshold for our FW gap measure are

$$\mathrm{tol} = \varepsilon \mathrm{SSE}_0 \text{ or } \mathrm{tol} = \varepsilon(n-1)\,\mathrm{tr}(\boldsymbol{S}_n) \tag{25}$$

(cf. Equation (21)) for some small $\varepsilon > 0$, say, $\varepsilon = 10^{-6}$, where the sample covariance $\boldsymbol{S}_n$ can readily be obtained from the data. Last but not least, on the strength of Theorem 8, the K-means algorithm is further guaranteed to terminate in no more than $\lceil \frac{\mathrm{SSE}_0}{\mathrm{tol}} \rceil$ steps. No such estimates were previously available.

5 SIMULATION STUDY

We want to illustrate the results of Section 4 with a simulation study. The major numerical challenge in studying the convergence rate of Algorithm 5 is the fact that the iteration becomes stationary. Since our results are *non-asymptotic*, the strategy is to study the numerical convergence in the preasymptotic regime. For a given dataset and initial seeds, the latter regime can be too short to facilitate any reasonable statistical analysis. Therefore, we rather focus on the *worst-case* performance over randomly selected seeds – for a fixed dataset or over multiple datasets sampled from a given population. Letting g_t^{WC} denote the worst-case (i.e.,

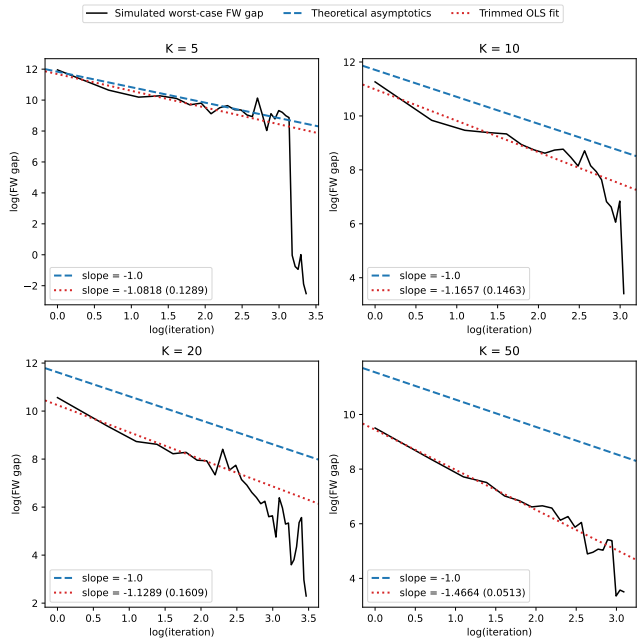

Figure 1: Worst-case convergence rate on synthetic "blobs" data for $n = 500$ and $d = 5$.

largest) FW gap (over all replications) at iteration t, from Figures 1 and 2, we observe that it tends to be monotonically decreasing in the preasymptotic regime so that we chose to study the later in lieu of the running minimum asymptotic gap $\tilde{g}_t$ appearing in Theorem 3. Empirically, the convergence rate of g_t^{WC} was estimated by applying OLS to the linear regression model

$$\log(g_t^{\mathrm{WC}}) = \beta_0 + \beta_1 \log(t+1) + \varepsilon$$

assuming i.i.d. normal errors ε. Preasymptotic regime was defined as the lower two thirds of the $\log(t)$ range. Theoretical values were assumed $\beta_1 = -1$ and $\beta_0 = (n-1)\,\mathrm{tr}(\boldsymbol{\Sigma})$, where the worst-case total variance was estimated empirically. All codes were implemented in plain Python and run in CPU mode on a 64-bit Ubuntu system on a Dell Precision 7960 Tower (Intel® Xeon(R) w7-3465X × 56 and 128 GB RAM).

Synthetic data. Consider the Gaussian mixture model $\sum_{k=1}^{K} \pi_k \varphi_k(\boldsymbol{x}|\boldsymbol{\mu}_k, \sigma_k^2 \boldsymbol{I})$ with equal proportions $\pi_k = \frac{1}{K}$ and equal spherical covariances with $\sigma_k = 1$ as implemented in `make_blobs()` of Python's `scikit-learn`. Whereas this is the usual assumption behind the K-means algorithm (Bishop, 2006, Chapter 9), real-world datasets typically depart from this model, i.e., the data could come from an elliptic mixture, not a spherical one. Aside from statistical implications, model violations tend to increase the runtime. Some authors even adopt uniformly distributed data (Hamerly, 2010) as a hypothetical worst-case scenario of "non-existing" cluster structure. Although we consider spherical mixtures, the clusters are rather poorly separated. Therefore, for small n, the data empirically look as if they were uniformly distributed.

For each (n, d, K) pair with $n = 500, 1000, 5000$, $d = 2, 5, 10$ and $K = 5, 10, 20, 50$, we performed 10,000 Monte Carlo (MC) simulations to estimate g_t^{WC} by randomly sampling both the data (from the 'blobs' mixture model) and initial seeds (from the dataset). The total runtime added up to about 11 hrs. Figure 1 illustrates one of the plots obtained, namely for $n = 500$ and $d = 5$. The discrepancy between the two lines appears to be smaller for lower values of K and d and larger values of n, suggesting improved ability to find better local minima in the latter case.

Figure 1 displays the worst-case (i.e., maximum over all replications) FW gap plot vs t in a log-log plot. Applying an upper-tailed Student's t-test of $H_0 : \beta_1 = -1$ vs $H_1 : \beta_1 > -1$ across all scenarios considered, the p-values never dropped below $1 - (1 - 0.05)^{1/36} \approx 0.0014$, which is the Šidák-adjusted limit to maintain a family-wise 0.05 test size, the composite null hypothesis failed to be rejected, corroborating that the worst-case preasymptotic rate was never slower than $\mathcal{O}(1/t)$. The estimated sloped are reported along with standard errors (in parentheses). The full set of plots as well as estimated slopes, standard errors and p-values are provided in Supplemental Sections B and C.

Image Segmentation Dataset. Consider the image segmentation dataset available from UCI Machine Learning Repository (Dua and Graff, 2025) recently studied by Pokojovy and Jobe (2022) in the context of robust estimation. Pooling training and test data, each of 2,310 instances is classified as one of the seven classes: `BRICKFACE`, `CEMENT`, `FOLIAGE`, `GRASS`, `PATH`, `SKY` and `WINDOW`. Keeping only continuous features (columns 6 through 19), we end up with 14 variables ($d = 14$). Pokojovy and Jobe (2022) empirically demonstrated that the 7-component mixture is non-Gaussian. Specifically, by analyzing the empirical quantiles of robust squared Mahalanobis distances of the `SKY` component, they showed that the former significantly exceeded respective χ_p^2-quantiles expected for Gaussian data. In addition to non-Gaussianity, the assumption of equal (not to mention spherical) covariances is also violated in this dataset. In the spirit of these observations, the image segmentation dataset is principally different from synthetic "blobs data" used in the previous simulation.

In a similar fashion, an MC simulation over 50,000 random seeds was performed for $K = 7$ with a total runtime of about 1 hr. The results are displayed in Figure 2. The slope estimate is reported along with the

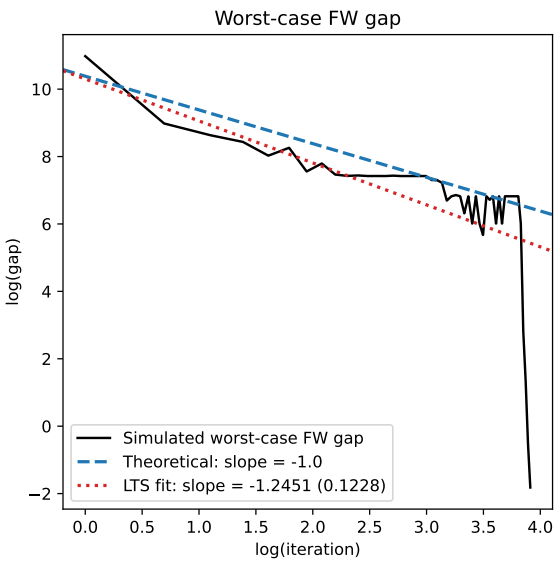

Figure 2: Worst-case convergence rate on image segmentation data.

standard error (in parentheses). Based on an upper-tailed Student's t-test with $\hat{\beta}_1 = -1.2451$ and $\mathrm{se}(\hat{\beta}_1) = 0.1228$, the null hypothesis $H_0 : \beta_1 = -1$ failed to be rejected against $H_1 : \beta_1 > -1$ at the test size of $\alpha = 0.05$, empirically corroborating an $\mathcal{O}(1/t)$ worst-case rate of Lloyd's K-means on this dataset.

Other Datasets. Other popular datasets (e.g., Fisher's Iris data, Swiss banknote data, etc.) as well as some datasets from Hamerly (2010) were analyzed in a fashion similar to the image segmentation dataset. The results were consistent with the ones reported above.

Computer Codes. The `Python` codes used to produce the empirical results reported in this work are available at `https://github.com/mpokojovy/kmeansFW`.

6 DISCUSSION & IMPLICATIONS

Discussion. We reiterate and emphasize the importance of non-asymptotic convergence guarantees for Lloyd's K-means – or any other machine learning algorithm for that matter – that hold in a uniform fashion with respect to the sample size, space dimension, hyperparameters, initial seeding or optimization "intricacies" like the curvature constant or geometric aspects of the domain. Building upon the recent precedent (Yurtsever and Sra, 2022) in the context of convex-concave procedure (CCCP), we proved that Lloyd's algorithm is a special case of FW. In doing so, we not only were able to perfectly carry over their results to Lloyd's K-means clustering but developed an FW variant for general semismooth concave objectives and proved an $\mathcal{O}(1/t)$ convergence rate. Owing to concavity, we could employ the less technical Clarke's instead of Goldstein's subdifferential prevalent in the nonsmooth literature allowing for better transparency and broader accessibility.

Practical Implications. By characterizing the drop in SSE at each step of the K-means algorithm as an FW gap and, thus, a type of Bregman divergence, we establish a new affine-invariant (Lacoste-Julien, 2016) alternative to existing error measures. Our measure allows for a simple stopping rule based on a natural choice of the tolerance threshold tol (cf. Equation (25)). Further, the maximum number of steps can be easily estimated. These discoveries can prove helpful both in traditional and emerging application domains, e.g., in federated multi-view clustering (Liu et al., 2023).

Extensions. With minor modifications, our results are applicable to trimmed K-means (Dorabiala et al., 2022; García-Escudero et al., 2008). As for the semismooth FW algorithm with concave objectives, an extension to infinite-dimensional Hilbert spaces is possible paving the way for optimal steering of dynamical systems via bang-bang control (Seyde et al., 2021).

Limitations. The limitations of our study are twofold. First, while our approach is applicable to a wide range of K-means variants and robust extensions, online and stochastic configurations are less obvious. The same applies to general elliptic mixtures that would require a better understanding of the negative trimmed log-likelihood-type objective (García-Escudero et al., 2008). Second, despite reassuring results, larger-scale simulations would be desirable in future investigations.

ACKNOWLEDGEMENTS

This work was partially supported by the National Science Foundation (DMS-2402544) and by the Canada CIFAR AI Chair Program. Simon Lacoste-Julien is a CIFAR Associate Fellow in the Learning in Machines & Brains Program.

References

Alayrac, J.-B., P. Bojanowski, N. Agrawal, I. Laptev, J. Sivic, and S. Lacoste-Julien (2016). Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4575–4583.

Arthur, D. and S. Vassilvitskii (2006). How slow is the k-means method? In *Proceedings of the Twenty-Second*

Annual Symposium on Computational Geometry, pp. 144–153. ACM.

Arthur, D. and S. Vassilvitskii (2007). k-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1027–1035. Society for Industrial and Applied Mathematics.

Bagirov, A. M. and E. Mohebi (2015). Nonsmooth optimization based algorithms in cluster analysis. In M. E. Celebi (Ed.), *Partitional Clustering Algorithms*, pp. 99–146. Springer.

Bauckhage, C. (2016). k-means clustering via the Frank-Wolfe algorithm. In *Proceedings of the LWDA 2016 Conference*, Volume 1670 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Bernholt, T. and P. Fischer (2004). The complexity of computing the MCD-estimator. *Theoretical Computer Science 326*(1-3), 383–398.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Information Science and Statistics. New York: Springer.

Blömer, J., C. Lammersen, M. Schmidt, and C. Sohler (2016). Theoretical analysis of the k-means algorithm – a survey. In *Algorithm Engineering*, Volume 9220 of *Lecture Notes in Computer Science*, pp. 81–116. Springer.

Bomze, I. M., F. Rinaldi, and S. R. Bulò (2019). First-order methods for the impatient: Support identification in finite time with convergent Frank–Wolfe variants. *SIAM Journal on Optimization 29*(3), 2211–2226.

Bomze, I. M., F. Rinaldi, and D. Zeffiro (2020). Active set complexity of the away-step Frank–Wolfe algorithm. *SIAM Journal on Optimization 30*(3), 2470–2500.

Bottou, L. and Y. Bengio (1995). Convergence properties of the k-means algorithms. In *Advances in Neural Information Processing Systems 7*, pp. 585–592. MIT Press.

Chari, V., S. Lacoste-Julien, I. Laptev, and J. Sivic (2015). On pairwise costs for network flow multi-object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5537–5545. IEEE.

Clarkson, K. L. (2010, September). Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. In *ACM Transactions on Algorithms (TALG)*, Volume 6, New York, NY, USA. Association for Computing Machinery.

de Oliveira, W. (2023). Short paper - A note on the Frank–Wolfe algorithm for a class of nonconvex and nonsmooth optimization problems. *Open Journal of Mathematical Optimization 4*(2), 1–10.

Dorabiala, O., J. N. Kutz, and A. Aravkin (2022). Robust trimmed k-means. *Pattern Recognition Letters 161*, 9–16.

Dua, D. and C. Graff (2025). UCI Machine Learning Repository. https://archive.ics.uci.edu/dataset/50/image+segmentation.

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

Frank, M. and P. Wolfe (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly 3*, 95–110.

Garber, D. and E. Hazan (2015). Faster rates for the Frank-Wolfe method over strongly-convex sets. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, Volume 37 of *ICML'15*, pp. 541–549. JMLR.

Garber, D. and E. Hazan (2016). A linearly convergent variant of the conditional gradient algorithm under strong convexity, with applications to online and stochastic optimization. *SIAM Journal on Optimization 26*(3), 1493–1528.

García-Escudero, L. A., A. Gordaliza, C. Matrán, and A. Mayo-Iscar (2008). A general trimming approach to robust cluster analysis. *The Annals of Statistics 36*(3), 1324–1345.

GuéLat, J. and P. Marcotte (1986). Some comments on Wolfe's 'away step'. *Mathematical Programming 35*(1), 110–119.

Hamerly, G. (2010). Making k-means even faster. In *Proceedings of the 2010 SIAM International Conference on Data Mining (SDM)*, pp. 130–140. Society for Industrial and Applied Mathematics.

Har-Peled, S. and A. Kushal (2007). Smaller coresets for k-median and k-means clustering. *Discrete & Computational Geometry 37*(1), 3–19.

Horst, R. (1984). On the global minimization of concave functions: Introduction and survey. *Operations Research Spektrum 6*(4), 195–205.

Ikotun, A. M., A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming (2023). k-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences 622*, 178–210.

Jaggi, M. (2013, 17–19 Jun). Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In S. Dasgupta and D. McAllester (Eds.), *Proceedings of the 30th International Conference on Machine Learning*,

Volume 28 of *Proceedings of Machine Learning Research*, Atlanta, Georgia, USA, pp. 427–435. PMLR.

Johnson, R. A. and D. W. Wichern (2007). *Applied Multivariate Statistical Analysis* (6th ed.). Upper River, NJ: Prentice Hall.

Khamaru, K. and M. J. Wainwright (2019). Convergence guarantees for a class of non-convex and non-smooth optimization problems. *Journal of Machine Learning Research 20*(154), 1–52.

Krishnan, R. G., S. Lacoste-Julien, and D. Sontag (2015). Barrier Frank-Wolfe for marginal inference. In *Advances in Neural Information Processing Systems (NeurIPS)*, Volume 28, pp. 532–540.

Kružík, M. (2000). Bauer's maximum principle and hulls of sets. *Calculus of Variations and Partial Differential Equations 11*(3), 321–332.

Lacoste-Julien, S. (2016). Convergence rate of Frank-Wolfe for non-convex objectives. arXiv:1607.00345.

Lacoste-Julien, S. and M. Jaggi (2015). On the global linear convergence of Frank-Wolfe optimization variants. In *Advances in Neural Information Processing Systems (NeurIPS)*, Volume 28.

Liu, J., X. Liu, E. Zhu, L. Liu, and J. Yin (2023). Communication-efficient federated multi-view clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence 45*(10), 12353–12369.

Liu, Z., C. Chen, L. Luo, and B. K. H. Low (2024). Zeroth-order methods for constrained nonconvex nonsmooth stochastic optimization. In *Proceedings of the 41st International Conference on Machine Learning*, Volume 235 of *Proceedings of Machine Learning Research*, pp. 30842–30872. PMLR.

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory 28*(2), 129–137.

Locatello, F., R. Khanna, M. Tschannen, and M. Jaggi (2017). A unified optimization view on generalized matching pursuit and Frank–Wolfe. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 860–868. PMLR.

MacKay, D. J. C. (2003). *Information Theory, Inference and Learning Algorithms*. Cambridge, UK: Cambridge University Press.

Mangasarian, O. L. (1996). Machine learning via polyhedral concave minimization. In H. Fischer, B. Riedmueller, and S. Schaeffler (Eds.), *Applied Mathematics and Parallel Computing—Festschrift for Klaus Ritter*, pp. 175–188. Heidelberg: Physica-Verlag.

Négiar, G., G. Dresdner, A. Y.-T. Tsai, L. El Ghaoui, F. Locatello, R. M. Freund, and F. Pedregosa (2020). Stochastic Frank–Wolfe for constrained finite-sum minimization. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, Volume 119 of *Proceedings of Machine Learning Research*, pp. 7253–7262. PMLR.

Nesterov, Y. E. (2018). *Lectures on Convex Optimization*, Volume 137 of *Optimization and Its Applications*. Berlin: Springer.

Pedregosa, F., G. Negiar, A. Askari, and M. Jaggi (2020, 26–28 Aug). Linearly convergent Frank-Wolfe with backtracking line-search. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, Volume 108 of *Proceedings of Machine Learning Research*, pp. 1–10. PMLR.

Pokojovy, M. and J. M. Jobe (2022). A robust deterministic affine-equivariant algorithm for multivariate location and scatter. *Computational Statistics & Data Analysis 172*(107475), 1–24.

Pokutta, S. (2023). The Frank-Wolfe algorithm: A short introduction. *Jahresbericht der Deutschen Mathematiker-Vereinigung 126*(3), 3–35.

Ravi, S. N., M. D. Collins, and V. Singh (2019). A deterministic nonsmooth Frank-Wolfe algorithm with coreset guarantees. *INFORMS Journal on Optimization 1*(2), 120–142.

Reddi, S. J., S. Sra, B. Póczós, and A. Smola (2016). Stochastic Frank-Wolfe methods for nonconvex optimization. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1244–1251. IEEE.

Rinaldi, F., F. Schoen, and M. Sciandrone (2009). Concave programming for minimizing the zero-norm over polyhedral sets. *Computational Optimization and Applications 43*(1), 1–22.

Rockafellar, R. T. and R. J.-B. Wets (1998). *Variational Analysis*, Volume 317 of *Grundlehren der Mathematischen Wissenschaften*. Berlin: Springer.

Selim, S. Z. and M. A. Ismail (1984). K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6*(1), 81–87.

Seyde, T., I. Gilitschenski, W. Schwarting, B. Stellato, M. Riedmiller, M. Wulfmeier, and D. Rus (2021). Is bang-bang control all you need? Solving continuous control with Bernoulli policies. In *Advances in Neural Information Processing Systems*, Volume 34, pp. 16337–16349.

Steinley, D. (2006). K-means clustering: A half-century synthesis. *British Journal of Mathematical and Statistical Psychology 59*(1), 1–34.

Thorup, M. (2004). Integer priority queues with decrease key in constant time and the single source shortest paths problem. *Journal of Computer and System Sciences 69*(3), 330–353.

White, D. J. (1993). Extension of the Frank-Wolfe algorithm to concave nondifferentiable objective functions. *Journal of Optimization Theory and Applications 78*(2), 283–301.

Wu, X., V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, and M. Steinbach (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems 14*(1), 1–37.

Yurtsever, A. and S. Sra (2022). CCCP is Frank-Wolfe in disguise. In *Advances in Neural Information Processing Systems (NeurIPS)*, Volume 35, pp. 29989–30001.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   Justification: The paper is mostly self-contained and provides a clear and thorough description of all mathematical settings, assumptions and algorithms. When necessary, specific references are provided for extra details.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   Justification: While FW is an abstract algorithm so that the complexity can only be measured with respect to the number of steps T, the usual complexity and storage hold for the K-means algorithm (see the paragraph below Algorithm 5 in Section 4).
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]
   Justification: Anonymized run-ready source code that solely relies on standard libraries is provided in the Supplement.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   Justification: All lemmas and theorems contain the full set of assumptions.
   (b) Complete proofs of all theoretical results. [Yes]
   Justification: Complete proofs of all theoretical results are provided in the paper and the Supplement.
   (c) Clear explanations of any assumptions. [Yes]
   Justification: All assumptions are clearly explained. Additional remarks are often provided to facilitate better understanding.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   Justification: All code, data and instructions (including seeds for random number generation) are included in the Supplement.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   Justification: This and other information is included in the supplemental code.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   Justification: Standard errors and p-values are reported and explained in Section 5 of the paper and the Supplement.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
   Justification: This information is provided in Section 5 of the paper.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   Justification: See Dua and Graff (2025).
   (b) The license information of the assets, if applicable. [Yes]
   Justification: Following the link provided (see Dua and Graff (2025)), the data are licensed under the Creative Commons Attribution 4.0 International (CC BY 4.0) license.
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   Justification: No new assets introduced.
   (d) Information about consent from data providers/curators. [Not Applicable]
   Justification: Not required under CC BY 4.0.
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
   Justification: The dataset is publicly available and does not include any sensible content.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   Justification: No participants were involved.
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   Justification: No participants were involved.
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]
   Justification: No participants were involved.

Supplementary Materials

A SUPPLEMENTARY PROOFS

A.1 Proof of Lemma 6

Proof. Observing

$$\partial_{w_{ik}} n_k^{\boldsymbol{w}} = 1, \quad \partial_{w_{ik}} \bar{\boldsymbol{x}}_k^{\boldsymbol{w}} = -\frac{1}{(n_k^{\boldsymbol{w}})^2} \sum_{i'=1}^{n} w_{i'k} \boldsymbol{x}_{i'} + \frac{1}{n_k^{\boldsymbol{w}}} \boldsymbol{x}_i = \frac{1}{n_k^{\boldsymbol{w}}} \big(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}\big),$$
$$\partial_{w_{ik}} \partial_{w_{jk}} \bar{\boldsymbol{x}}_k^{\boldsymbol{w}} = -\frac{1}{(n_k^{\boldsymbol{w}})^2} \big(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}\big) - \frac{1}{n_k^{\boldsymbol{w}}} \partial_{w_{jk}} \bar{\boldsymbol{x}}_k^{\boldsymbol{w}} = -\frac{1}{n_k^{\boldsymbol{w}}} \Big(\big(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}\big) + \big(\boldsymbol{x}_j - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}\big) \Big),$$

we compute the first-order

$$\begin{aligned} \partial_{w_{ik}} f(\boldsymbol{w}) &= \partial_{w_{ik}} \Big(\sum_{k'=1}^{K} \sum_{i'=1}^{n} w_{i'k'} \|\boldsymbol{x}_{i'} - \bar{\boldsymbol{x}}_{k'}^{\boldsymbol{w}}\|^2 \Big) \\ &= \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}\|^2 - \frac{2}{n_k^{\boldsymbol{w}}} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}) \cdot \Big(\sum_{i'=1}^{n} w_{i'k} (\boldsymbol{x}_{i'} - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}) \Big) = \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}\|^2 \end{aligned}$$

and the second-order derivatives

$$\partial_{w_{ik}} \partial_{w_{jl}} f(\boldsymbol{w}) = -2(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}) \cdot \partial_{w_{jl}} \bar{\boldsymbol{x}}_k^{\boldsymbol{w}} = -\frac{2}{n_k^{\boldsymbol{w}}} \cdot \mathbb{1}_{\{k=l\}} \cdot (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}) \cdot (\boldsymbol{x}_j - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}).$$

For $\boldsymbol{\xi} \in \mathbb{R}^{n \times K}$, we can write

$$\begin{aligned} \big\langle \nabla^2 f(\boldsymbol{w}) \boldsymbol{\xi}, \boldsymbol{\xi} \big\rangle_{\mathcal{F}} &= \sum_{k=1}^{K} \sum_{i=1}^{n} \xi_{ik} \sum_{i'=1}^{n} \Big(-\frac{2}{n_k^{\boldsymbol{w}}} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}) \cdot (\boldsymbol{x}_{i'} - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}) \Big) \xi_{i'k} \\ &= -2 \sum_{k=1}^{K} \frac{1}{n_k^{\boldsymbol{w}}} \sum_{i=1}^{n} \sum_{i'=1}^{n} \xi_{ik} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}) \cdot (\boldsymbol{x}_{i'} - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}) \xi_{i'k} \\ &= -2 \sum_{k=1}^{K} \frac{1}{n_k^{\boldsymbol{w}}} \Big(\sum_{i=1}^{n} \xi_{ik} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}) \Big) \cdot \Big(\sum_{i'=1}^{n} \xi_{ik} (\boldsymbol{x}_{i'} - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}) \Big) \\ &= -2 \sum_{k=1}^{K} \frac{1}{n_k^{\boldsymbol{w}}} \Big\| \sum_{i=1}^{n} \xi_{ik} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k^{\boldsymbol{w}}) \Big\|^2. \end{aligned}$$

Thus, $\big\langle \nabla^2 f(\boldsymbol{w}) \boldsymbol{\xi}, \boldsymbol{\xi} \big\rangle_{\mathcal{F}} \leq 0$. □

A.2 Proof of Remark 7

Proof. We compute

$$\begin{aligned}\Big|\big\langle \nabla^2 f(\boldsymbol{w})\boldsymbol{\xi},\boldsymbol{\xi}\big\rangle_{\mathcal{F}}\Big| &\leq 2\sum_{k=1}^{K}\frac{1}{n_k^{\boldsymbol{w}}}\Big(\max_{\substack{i=1,\dots,n\\k=1,\dots,K}}\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_k^{\boldsymbol{w}}\|\Big)^2\sum_{i=1}^{n}\xi_{ik}^2\\ &\leq 2\Big(\max_{i,j=1,\dots,n}\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2\Big)\|\boldsymbol{\xi}\|_{\mathcal{F}}^2\\ &\leq 2\|\boldsymbol{\xi}\|_{\mathcal{F}}^2\max_{\substack{i=1,\dots,n\\k=1,\dots,K}}\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_k^{\boldsymbol{w}}\|^2\\ &\leq 2\max_{\substack{i=1,\dots,n\\k=1,\dots,K}}\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_k^{\boldsymbol{w}}\|^2\leq 2\Big(\max_{i,j=1,\dots,n}\|\boldsymbol{x}_i-\boldsymbol{x}_j\|\Big)^2\end{aligned}$$

for all $\boldsymbol{\xi}\in\mathbb{R}^{n\times K}$, which estimates the operator norm of $\nabla^2 f$. □

B ADDITIONAL FIGURES

B.1 Dimension $d = 2$

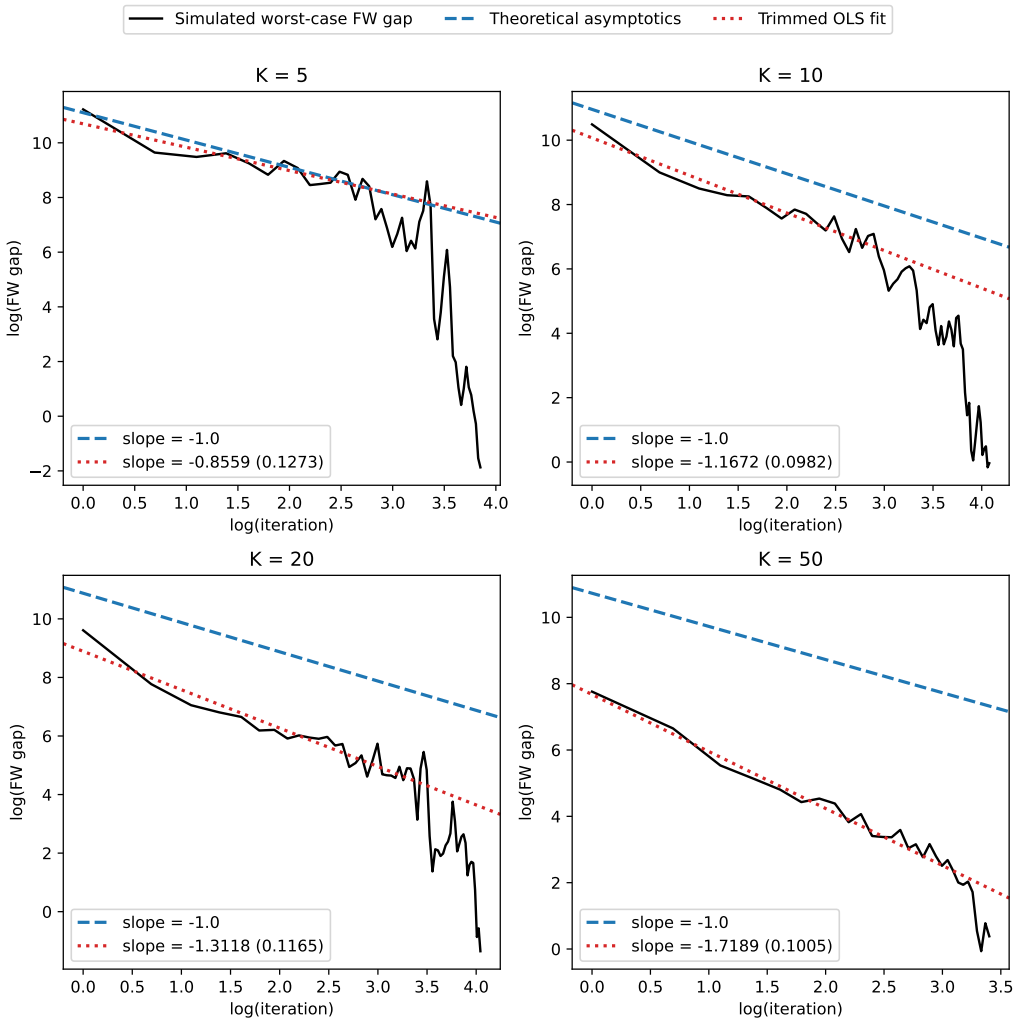

Figure 3: Worst-case FW convergence rate on synthetic "blobs" data for $n = 500$ and $d = 2$.

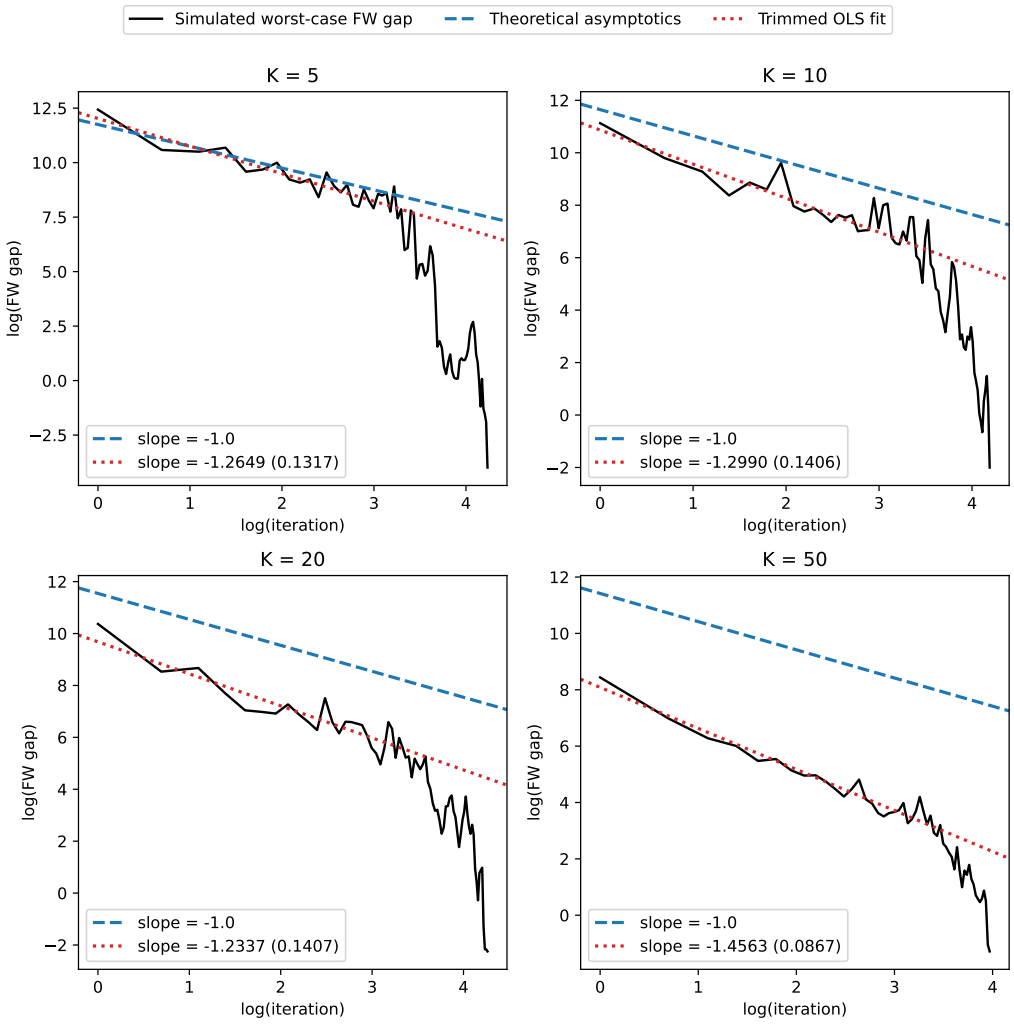

Figure 4: Worst-case FW convergence rate on synthetic "blobs" data for $n = 1000$ and $d = 2$.

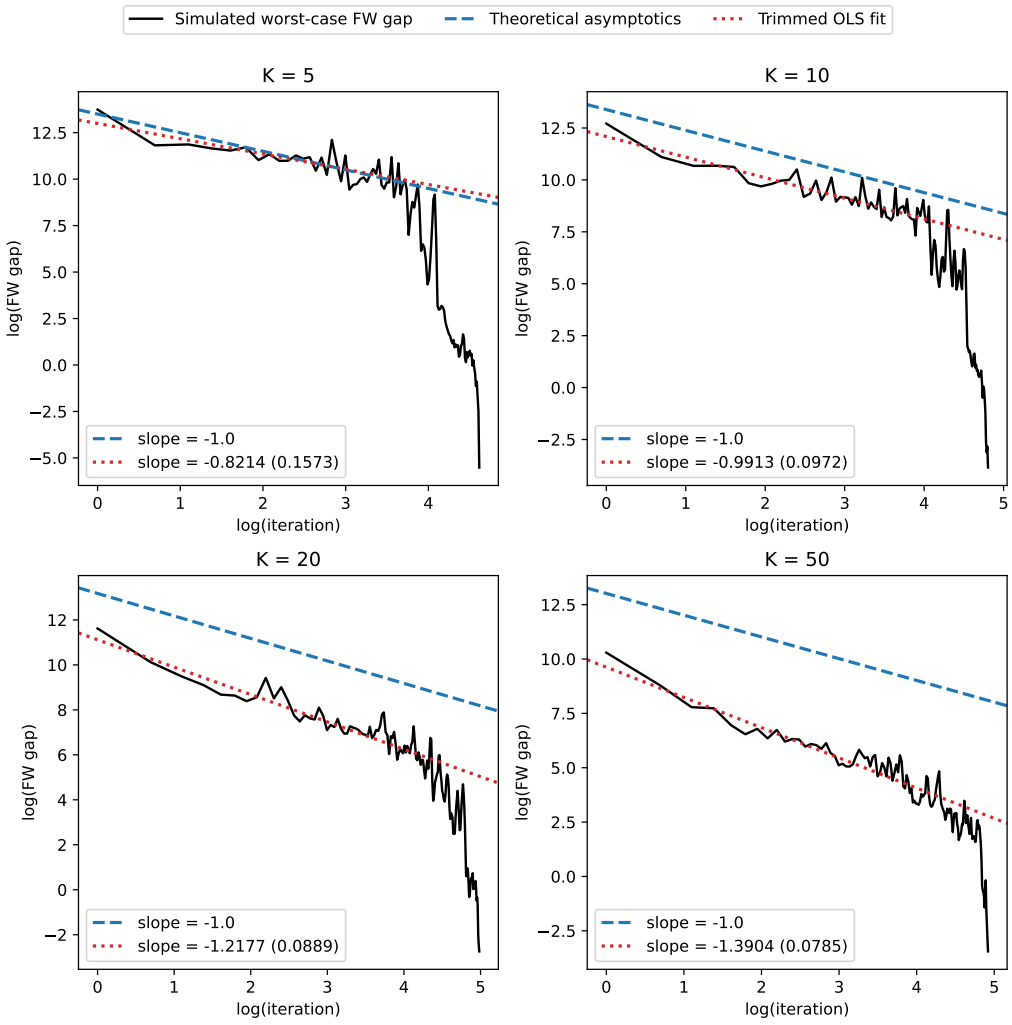

Figure 5: Worst-case FW convergence rate on synthetic "blobs" data for $n = 5000$ and $d = 2$.

B.2 Dimension $d = 5$

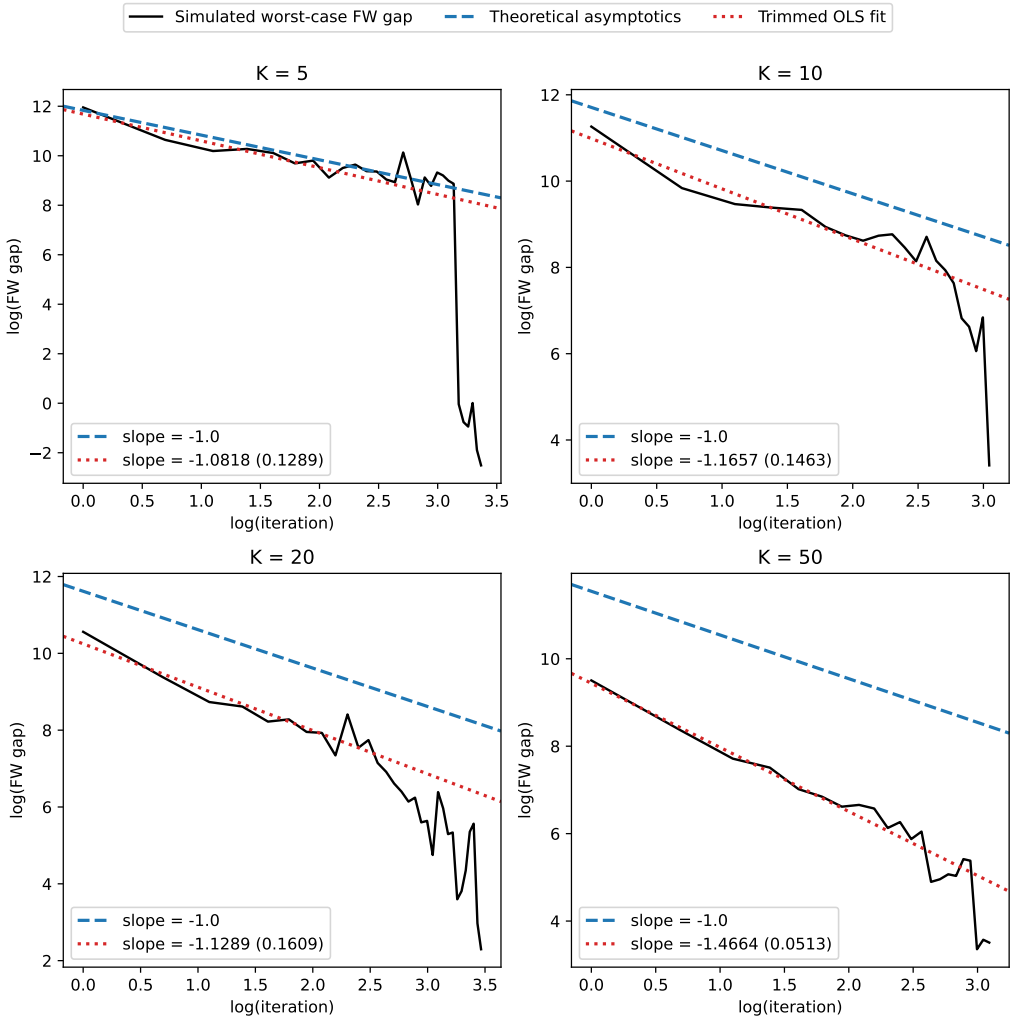

Figure 6: Worst-case FW convergence rate on synthetic "blobs" data for $n = 500$ and $d = 5$.

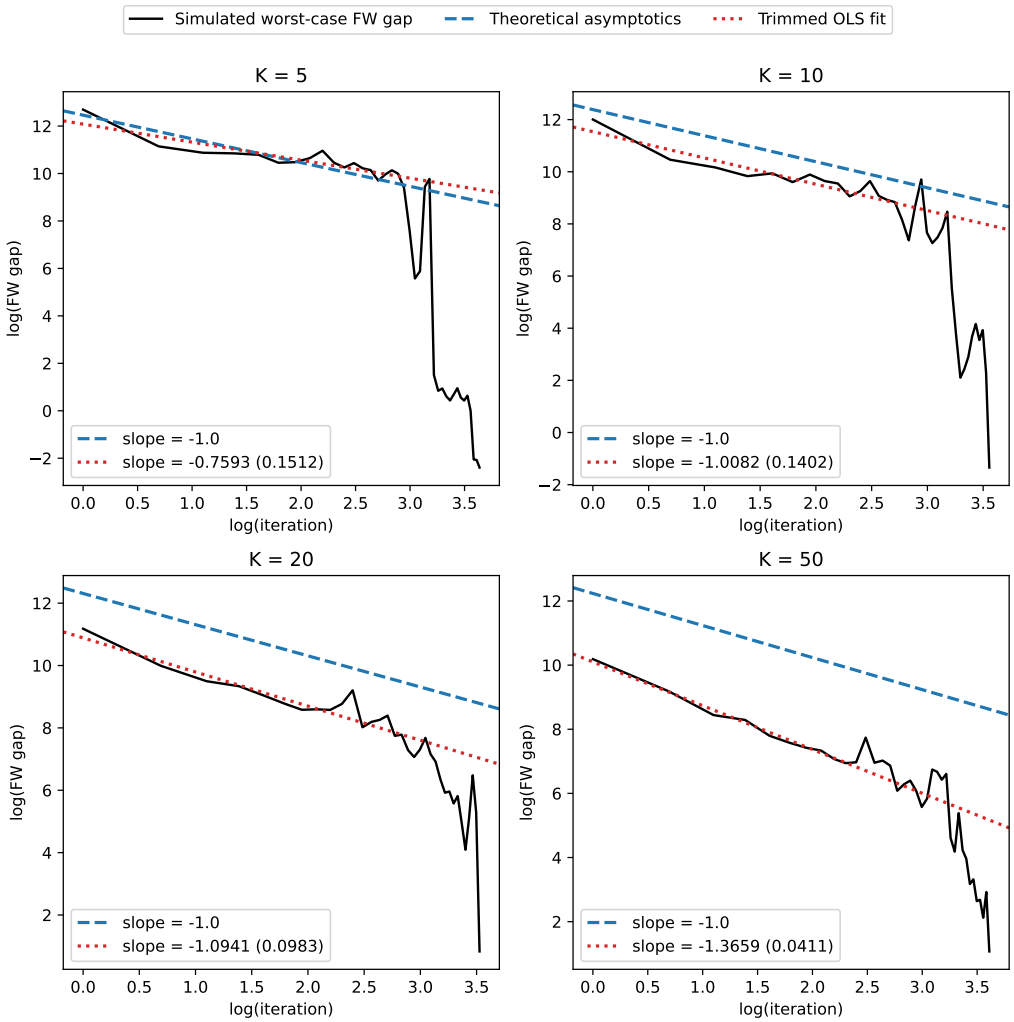

Figure 7: Worst-case FW convergence rate on synthetic "blobs" data for $n = 1000$ and $d = 5$.

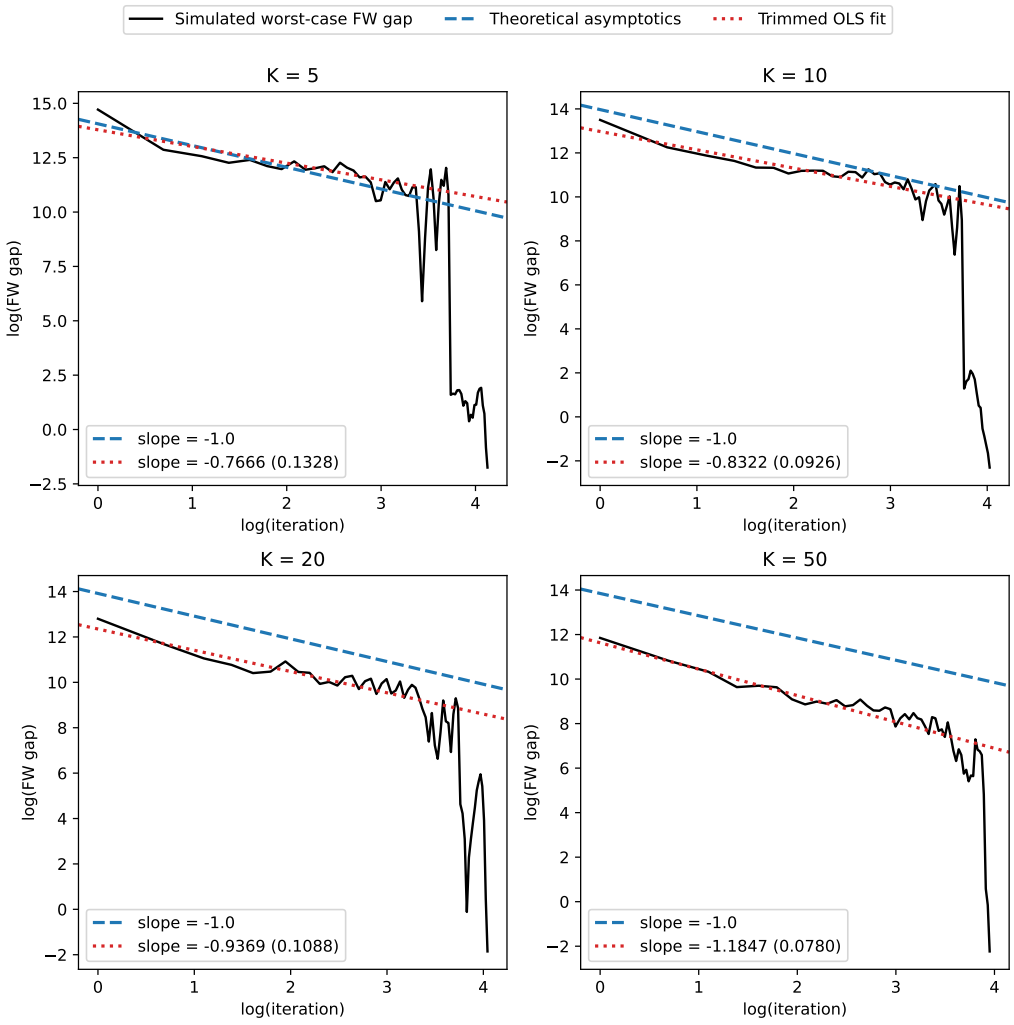

Figure 8: Worst-case FW convergence rate on synthetic "blobs" data for $n = 5000$ and $d = 5$.

B.3 Dimension $d = 10$

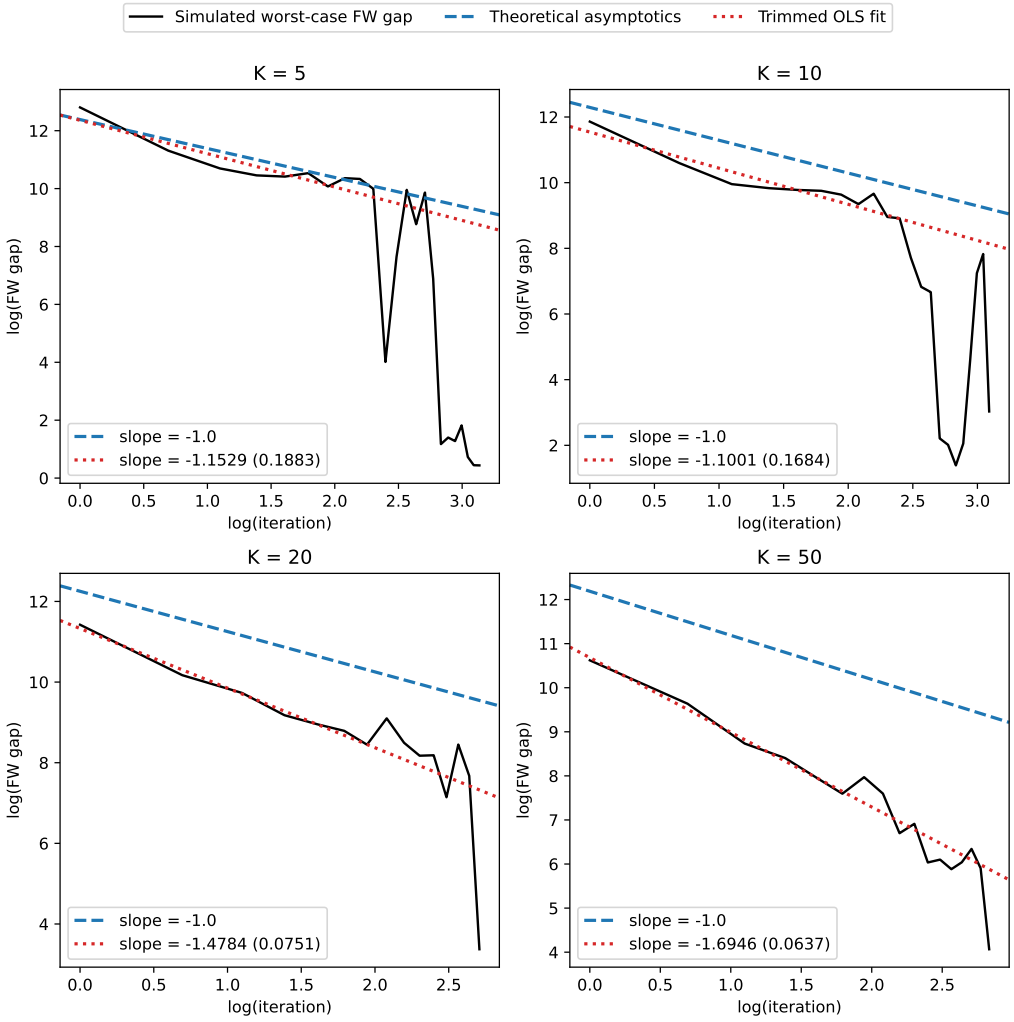

Figure 9: Worst-case FW convergence rate on synthetic "blobs" data for $n = 500$ and $d = 10$.

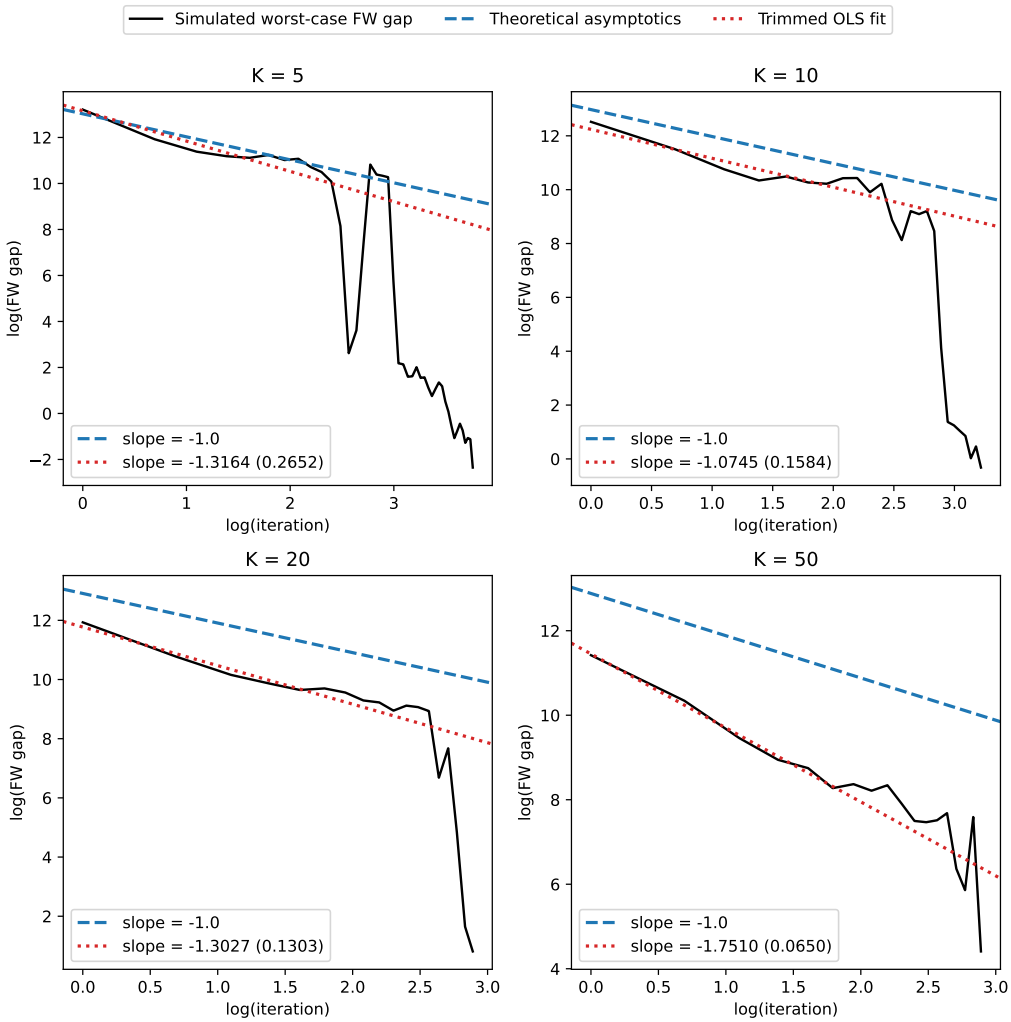

Figure 10: Worst-case FW convergence rate on synthetic "blobs" data for $n = 1000$ and $d = 10$.

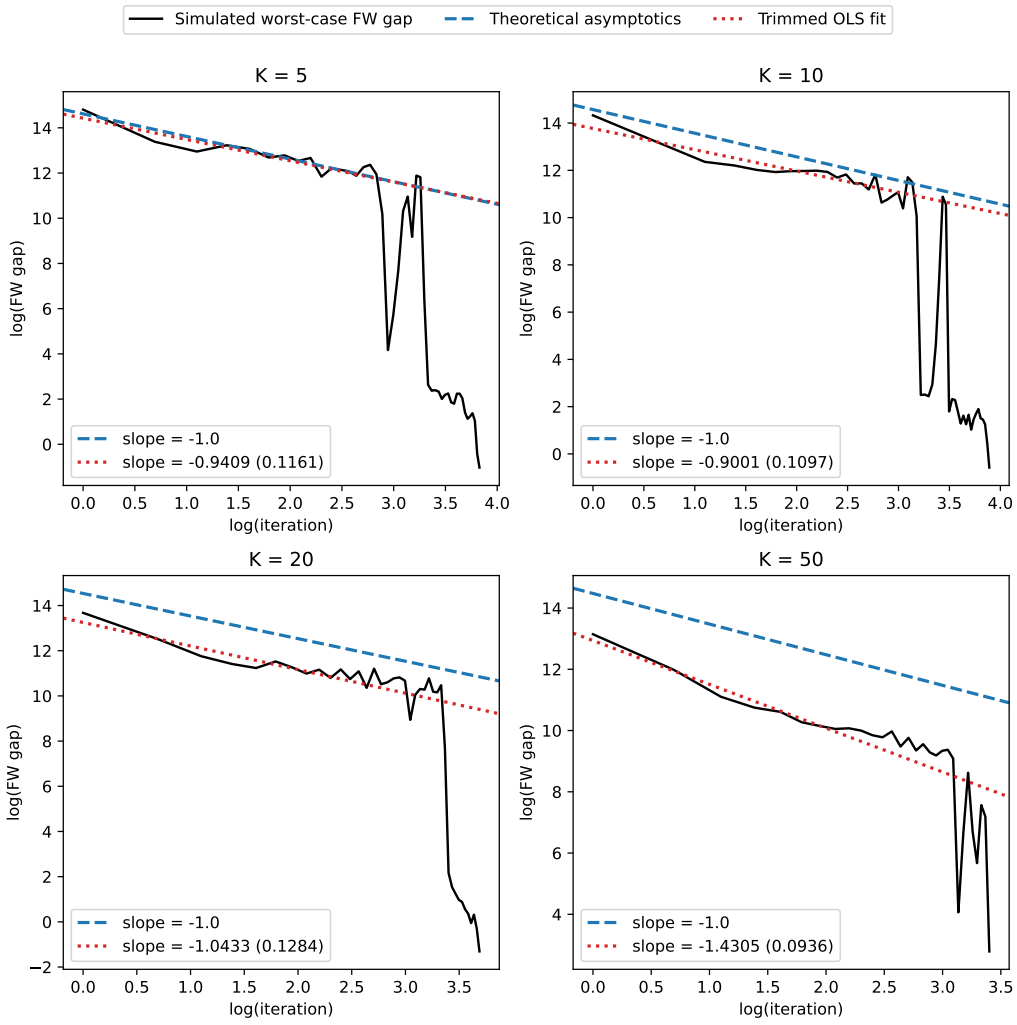

Figure 11: Worst-case FW convergence rate on synthetic "blobs" data for $n = 5000$ and $d = 10$.

C TABULATED SIMULATION RESULTS

The p-values reported below are from Student's t-test of $H_0 : \beta_1 = -1$ vs $H_1 : \beta_1 > -1$.

Table 1: Simulation results for $d = 2$.

		Theoretical bound		Empirical fit		
n	K	Slope	Intercept	Slope (SE)	Intercept (SE)	p-value
500	5	-1.0000	11.1017	-0.8559 (0.1273)	10.6940 (0.2399)	0.1409
500	10	-1.0000	10.9562	-1.1672 (0.0982)	10.0712 (0.1971)	0.9438
500	20	-1.0000	10.8755	-1.3118 (0.1165)	8.8922 (0.2268)	0.9899
500	50	-1.0000	10.7266	-1.7189 (0.1005)	7.6737 (0.1584)	0.9999
1000	5	-1.0000	11.7480	-1.2649 (0.1317)	12.0234 (0.2717)	0.9681
1000	10	-1.0000	11.6464	-1.2990 (0.1406)	10.8671 (0.2902)	0.9741
1000	20	-1.0000	11.5446	-1.2337 (0.1407)	9.6810 (0.2978)	0.9413
1000	50	-1.0000	11.4175	-1.4563 (0.0867)	8.0852 (0.1689)	0.9999
5000	5	-1.0000	13.4988	-0.8214 (0.1573)	12.9955 (0.3623)	0.1351
5000	10	-1.0000	13.3859	-0.9913 (0.0972)	12.0877 (0.2354)	0.4645
5000	20	-1.0000	13.1775	-1.2177 (0.0889)	11.1182 (0.2250)	0.9891
5000	50	-1.0000	13.0158	-1.3904 (0.0785)	9.6238 (0.1960)	1.0000

Table 2: Simulation results for $d = 5$.

		Theoretical bound		Empirical fit		
n	K	Slope	Intercept	Slope (SE)	Intercept (SE)	p-value
500	5	-1.0000	11.8339	-1.0818 (0.1289)	11.6820 (0.2031)	0.7272
500	10	-1.0000	11.7078	-1.1657 (0.1463)	10.9860 (0.2009)	0.8455
500	20	-1.0000	11.6158	-1.1289 (0.1609)	10.2478 (0.2676)	0.7769
500	50	-1.0000	11.5441	-1.4664 (0.0513)	9.4392 (0.0705)	0.9999
1000	5	-1.0000	12.4600	-0.7593 (0.1512)	12.0811 (0.2634)	0.0729
1000	10	-1.0000	12.3841	-1.0082 (0.1402)	11.5376 (0.2332)	0.5227
1000	20	-1.0000	12.3113	-1.0941 (0.0983)	10.8865 (0.1635)	0.8167
1000	50	-1.0000	12.2349	-1.3659 (0.0411)	10.1017 (0.0716)	1.0000
5000	5	-1.0000	14.0585	-0.7666 (0.1328)	13.7828 (0.2666)	0.0512
5000	10	-1.0000	13.9688	-0.8322 (0.0926)	12.9732 (0.1804)	0.0476
5000	20	-1.0000	13.9161	-0.9369 (0.1088)	12.3498 (0.2120)	0.2865
5000	50	-1.0000	13.8460	-1.1847 (0.0780)	11.6294 (0.1470)	0.9814

Table 3: Simulation results for $d = 10$.

		Theoretical bound		Empirical fit		
n	K	Slope	Intercept	Slope (SE)	Intercept (SE)	p-value
500	5	-1.0000	12.3848	-1.1529 (0.1883)	12.3593 (0.2787)	0.7761
500	10	-1.0000	12.2919	-1.1001 (0.1684)	11.5409 (0.2313)	0.7109
500	20	-1.0000	12.2521	-1.4784 (0.0751)	11.3285 (0.0940)	0.9984
500	50	-1.0000	12.1881	-1.6946 (0.0637)	10.6826 (0.0798)	0.9998
1000	5	-1.0000	13.0277	-1.3164 (0.2652)	13.1580 (0.4815)	0.8698
1000	10	-1.0000	12.9742	-1.0745 (0.1584)	12.2400 (0.2344)	0.6727
1000	20	-1.0000	12.9085	-1.3027 (0.1303)	11.7727 (0.1631)	0.9596
1000	50	-1.0000	12.8820	-1.7510 (0.0650)	11.4515 (0.0814)	0.9998
5000	5	-1.0000	14.6173	-0.9409 (0.1161)	14.4244 (0.2108)	0.3109
5000	10	-1.0000	14.5702	-0.9001 (0.1097)	13.7718 (0.2067)	0.1909
5000	20	-1.0000	14.5348	-1.0433 (0.1284)	13.2512 (0.2238)	0.6280
5000	50	-1.0000	14.4765	-1.4305 (0.0936)	12.9385 (0.1475)	0.9988