Frank-Wolfe Optimization for Symmetric-NMF under Simplicial Constraint

Han Zhao
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213
han.zhao@cs.cmu.edu

Geoff Gordon
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213
ggordon@cs.cmu.edu

Abstract

Symmetric nonnegative matrix factorization has found abundant applications in various domains by providing a symmetric low-rank decomposition of nonnegative matrices. In this paper we propose a Frank-Wolfe (FW) solver to optimize the symmetric nonnegative matrix factorization problem under a simplicial constraint, which has recently been proposed for probabilistic clustering. Compared with existing solutions, this algorithm is simple to implement, and has no hyperparameters to be tuned. Building on the recent advances of FW algorithms in nonconvex optimization, we prove an $O(1/\varepsilon^2)$ convergence rate to ε-approximate KKT points, via a tight bound $\Theta(n^2)$ on the curvature constant, which matches the best known result in unconstrained nonconvex setting using gradient methods. Numerical results demonstrate the effectiveness of our algorithm. As a side contribution, we construct a simple nonsmooth convex problem where the FW algorithm fails to converge to the optimum. This result raises an interesting question about necessary conditions of the success of the FW algorithm on convex problems.

1 INTRODUCTION

Nonnegative matrix factorization (NMF) has found various applications in data mining, natural language processing, and computer vision [Xu et al., 2003, Liu et al., 2003, Kuang et al., 2012], due to its ability to provide low rank approximations and interpretable decompositions. The decision version of NMF is known to be NP-hard [Vavasis, 2009], which also implies that its optimization problem is NP-hard. Recently, a variant of NMF where the input matrix is constrained to be symmetric has become popular for clustering [He et al., 2011, Kuang et al., 2012, 2015]. The problem is known as symmetric NMF (SymNMF), and its goal is to minimize $||A - WW^T||_F^2$ under the constraint that $W \geq 0$ elementwise, where A is a symmetric matrix of cluster affinities. Compared with NMF, SymNMF is applicable even when the algorithm does not have direct access to the data instances, but only their pairwise similarity scores. Note that in general the input matrix A does not need to be nonnegative [Kuang et al., 2012]. SymNMF has been successfully applied in many different settings and was shown to be competitive with standard clustering algorithms; see [Kuang et al., 2012, 2015] and the references therein for more details.

In this paper we investigate a constrained version of SymNMF where the input matrix is required to be both nonnegative and positive semidefinite. Furthermore, we require that W is normalized such that each row of W sums to 1. This problem has an interesting application in probabilistic clustering [Zhao et al., 2015] where the ith row of W can be interpreted as the probability that the ith data point lies in each clusters. Formally, we are interested in the following optimization problem, which we name as simplicial SymNMF (SSymNMF):

$$
\begin{aligned}
& \underset{W}{\text{minimize}} & & \frac{1}{4}||P - WW^T||_F^2 \\
& \text{subject to} & & W \in \mathbb{R}_+^{n \times k}, \quad W\mathbf{1}_k = \mathbf{1}_n
\end{aligned}
\tag{1}
$$

where $P \geq 0$ and $P \in \mathbb{S}_+^n$ is positive semidefinite. (1) was proposed as a formulation for probabilistic clustering [Zhao et al., 2015]. The input matrix P is interpreted as the co-cluster affinity matrix, i.e., entry P_{ij} corresponds to the degree to which we encourage data instances $\mathbf{x}_i$ and $\mathbf{x}_j$ to be in the same cluster. Each row of W then corresponds to the probability distribution of instance $\mathbf{x}_i$ being in different clusters. A similar simplicial constraint has been considered in NMF as well [Nguyen et al., 2013], where the goal is to seek a probabilistic part-based decomposition for clustering.

Previous approaches to solve SymNMF or SSymNMF use the penalty method to convert it into an unconstrained optimization problem and then solve it iteratively, using either first-order or second-order methods [Kuang et al., 2012, Zhao et al., 2015, Kuang et al., 2015]. Such methods usually include two loops where the outer loop gradually increases the penalty coefficients and the inner loop finds a stationary point of each fixed penalized objective, hence they are often computationally expensive and slow to converge. In this paper we first give an equivalent geometric description of (1) and then propose a variant of the classic Frank-Wolfe (FW) algorithm [Frank and Wolfe, 1956], a.k.a. the conditional gradient method [Levitin and Polyak, 1966], to solve it. We also provide a non-asymptotic convergence guarantee of our algorithm under an affine invariant stationarity measure (defined in Sec. 2). More specifically, for a given approximation parameter $\varepsilon > 0$, we show that the algorithm converges to an ε-approximate KKT point of (1) in $O(1/\varepsilon^2)$ iterations. This rate is analogous to the one derived by Nesterov [2013] for general unconstrained problems (potentially nonconvex) using the gradient descent method, where the measure of stationarity is given by the norm of the gradient. The $O(1/\varepsilon^2)$ rate has recently been shown to be optimal [Cartis et al., 2010] in the unconstrained setting for gradient methods, and it also matches the best known rate to a stationary point with (accelerated) projected gradient descent [Ghadimi and Lan, 2016, Ghadimi et al., 2016] in the constrained smooth nonconvex setting.

Contributions. We first give a generalized definition of the curvature constant [Jaggi, 2013, Lacoste-Julien et al., 2013, Lacoste-Julien, 2016] that works for both convex and nonconvex functions, and we prove a tight bound of it in (1). We then propose a convergence measure in terms of the duality gap and show that the gap is 0 iff KKT conditions are satisfied. Using these two tools, we propose a FW algorithm to solve (1) and show that it has a non-asymptotic convergence rate $O(1/\varepsilon^2)$ to KKT points. On the algorithmic side, we give a procedure that has the optimal linear time complexity and constant space complexity to implement the *linear minimization oracle* (LMO) in the FW algorithm. As a side contribution, we construct a piecewise linear example where the FW algorithm fails to converge to the optimum. Surprisingly, we can also show that the FW algorithm works if we slightly change the objective function, despite that the new function remains piecewise linear and has an unbounded curvature constant. These two examples then raise an interesting question w.r.t. the necessary condition of the success of the FW algorithm. At the end, we conduct several numerical experiments to demonstrate the efficiency of the proposed algorithm by comparing it with the penalty method and projected gradient descent.

2 PRELIMINARY

2.1 SymNMF UNDER SIMPLICIAL CONSTRAINT

One way to understand (1) is through its clustering based explanation: the goal is to find a probabilistic clustering of all the instances such that the given co-cluster affinity P_{ij} for a pair of instances $(\mathbf{x}_i, \mathbf{x}_j)$ is close to the true probability that $\mathbf{x}_i$ and $\mathbf{x}_j$ reside in the same cluster:

$$
\begin{aligned}
\Pr(\mathbf{x}_i \sim \mathbf{x}_j) &= \sum_{h=1}^{k} \Pr(c_i = c_j = h) \\
&= \sum_{h=1}^{k} \Pr(c_i = h) \Pr(c_j = h) = \mathbf{w}_i^T \mathbf{w}_j
\end{aligned}
$$

where we use the notation $\mathbf{x}_i \sim \mathbf{x}_j$ to mean "$\mathbf{x}_i$ and $\mathbf{x}_j$ reside in the same cluster"; c_i is the cluster assignment of $\mathbf{x}_i$ and $\mathbf{w}_i$ denotes the ith row vector of W. Note that the second equation holds because of the assumption of i.i.d. generation process of instances and their cluster assignments.

As a first note, the optimal solution W^* to (1) is not unique: for any permutation π_n over $[n]$, an equivalent solution can be constructed by $W^*_{\pi_n} = W^* \Pi_{\pi_n}$, where Π_{π_n} is a permutation matrix specified by π_n. This corresponds to an equivalence class of W by label switching. Hence for any fixed k, there are at least $k!$ optimal solutions to (1). The uniqueness of the solution to (1) up to permutation is still an open problem. Huang et al. [2014] studied sufficient and necessary conditions for the uniqueness of SymNMF, but they are NP-hard to check in general.

Zhao et al. [2015] proposed a penalty method to transform (1) into an unconstrained problem and solve it via sequential minimization. Roughly speaking, the penalty method repeatedly solves an unconstrained problem, and enforces the constraints in (1) by gradually increasing the coefficients of the penalty terms. This process iterates until a solution is both feasible and a stopping criterion w.r.t. the objective function is met; see [Zhao et al., 2015, Algo. 1]. The penalty method contains 6 different hyperparameters to be tuned, and it is not even clear whether it will converge to a KKT point of (1). To the best of our knowledge, no other methods has been proposed to solve (1). To solve SymNMF, Kuang et al. [2012] proposed a projected Newton method and Vandaele et al. [2016] developed a block coordinate descent method. However, due to the coupling of columns of W introduced by the simplicial constraint, it is not clear how to extend these two algorithms to solve (1). On the other hand, the simplicial constraint in (1) restricts the feasible set to be compact, which makes it possible for us to apply the FW algorithm to solve it.

2.2 Frank-Wolfe ALGORITHM

The FW algorithm [Frank and Wolfe, 1956, Levitin and Polyak, 1966] is a popular first-order method to solve constrained convex optimization problems of the following form:

$$\text{minimize}_{\mathbf{x} \in \mathcal{D}} \quad f(\mathbf{x}) \tag{2}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is a convex and continuously differentiable function over the convex and compact domain $\mathcal{D}$. The FW method has recently attracted a surge of interest in machine learning due to the fact that it never requires us to project onto the constraint set $\mathcal{D}$ and its ability to cheaply exploit structure in $\mathcal{D}$ [Clarkson, 2010, Jaggi, 2011, 2013, Lacoste-Julien et al., 2013, Lan, 2013]. Compared with projected gradient descent, it provides arguably wider applicability since projection can often be computationally expensive (e.g., for the general ℓ_p ball), and in some case even computationally intractable [Collins et al., 2008]. At each iteration, the FW algorithm finds a feasible search corner $\mathbf{s}$ by minimizing a linear approximation at the current iterate $\mathbf{x}$ over $\mathcal{D}$:

$$w(\mathbf{x}) := \min_{\hat{\mathbf{s}} \in \mathcal{D}} f(x) + \nabla f(\mathbf{x})^T (\hat{\mathbf{s}} - \mathbf{x}) \tag{3}$$

The linear minimization oracle (LMO) at $\mathbf{x}$ is defined as:

$$\mathbf{s} = \text{LMO}(\mathbf{x}) := \arg\min_{\hat{\mathbf{s}} \in \mathcal{D}} f(x) + \nabla f(\mathbf{x})^T (\hat{\mathbf{s}} - \mathbf{x}) \tag{4}$$

Given $\mathbf{s} = \text{LMO}(\mathbf{x})$, the next iterate is then updated as a convex combination of $\mathbf{s}$ and the current iterate $\mathbf{x}$. If the FW algorithm starts with a corner as the initial point, then this property implies that at the t-th iteration, the current iterate is a convex combination of at most $t + 1$ corners. The difference between $f(\mathbf{x})$ and $w(x)$ is known as the *Frank-Wolfe gap* (FW-gap):

$$g(\mathbf{x}) := f(\mathbf{x}) - w(\mathbf{x}) = \max_{\hat{\mathbf{s}} \in \mathcal{D}} \nabla f(\mathbf{x})^T (\mathbf{x} - \hat{\mathbf{s}}) \tag{5}$$

which turns out to be a special case of the general Fenchel duality gap when we transform (2) into an unconstrained problem by adding an indicator function over $\mathcal{D}$ into the objective function [Lacoste-Julien et al., 2013, Appendix D]. Due to the convexity of f, we have the following inequality: $\forall \mathbf{x}, \mathbf{y} \in \mathcal{D}, \quad w(\mathbf{y}) \leq f^* \leq f(\mathbf{x})$, where f^* is the globally optimal value of f. Specifically, $\forall \mathbf{x} \in \mathcal{D}$, $w(\mathbf{x}) \leq f(\mathbf{x})$ and as a result $g(\mathbf{x})$ can be used as an upper bound for the optimality gap: $\forall \mathbf{x} \in \mathcal{D}, \quad f(\mathbf{x}) - f^* \leq g(\mathbf{x})$, so that $g(\mathbf{x})$ is a certificate for the approximation quality of the current solution. Furthermore, the duality gap $g(\mathbf{x})$ can be computed *essentially for free* in each iteration: $g(\mathbf{x}) = \nabla f(\mathbf{x})^T (\mathbf{x} - \mathbf{s})$.

It is well known that for smooth convex optimization problems the FW algorithm converges to an ε-approximate solution in $O(1/\varepsilon)$ iterations [Frank and Wolfe, 1956,

Dunn and Harshbarger, 1978]. This result has recently been generalized to the setting where the LMO is solved only approximately [Clarkson, 2010, Jaggi, 2013]. The analysis of convergence depends on a crucial concept known as the *curvature constant C_f* defined for a convex function f:

$$C_f := \sup_{\substack{\mathbf{x}, \mathbf{s} \in \mathcal{D}, \\ \gamma \in (0,1], \\ \mathbf{y} = \mathbf{x} + \gamma(\mathbf{s} - \mathbf{x})}} \frac{2}{\gamma^2} \left(f(\mathbf{y}) - f(\mathbf{x}) - \nabla f(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) \right) \tag{6}$$

The curvature constant measures the relative deviation of a convex function f from its linear approximation. It is clear that $C_f \geq 0$. Furthermore, the definition of C_f only depends on the inner product of the underlying space, which makes it affine invariant. In fact, the curvature constant, the FW algorithm, and its convergence analysis are all affine invariant [Lacoste-Julien et al., 2013]. Later we shall generalize the curvature constant to a function f that is not necessarily convex and state our result in terms of the generalized definition. The above definition of the curvature constant still works for nonconvex functions, but for a concave function f, $C_f = 0$, which loses its geometric interpretation as measuring the curvature of f.

To proceed our discussion, we first establish some standard terminologies used in this paper. A continuously differentiable function f is called L-smooth w.r.t. the norm $||\cdot||$ if ∇f is L-Lipschitz continuous w.r.t. the norm $||\cdot||$: $\forall \mathbf{x}, \mathbf{y} \in \mathcal{D}, \quad ||\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})|| \leq L||\mathbf{x} - \mathbf{y}||$. Throughout this paper, we will use $||\cdot||$ to mean the Euclidean norm, i.e., the ℓ_2 norm for vectors and the Frobenius norm for matrices if not explicitly specified. It is standard to assume f to be L-smooth in the convergence analysis of the FW algorithm [Clarkson, 2010, Jaggi, 2013]. As we will see below, the smoothness of f also implies the boundedness of C_f on a compact set.

3 Frank-Wolfe FOR SymNMF UNDER SIMPLICIAL CONSTRAINT

3.1 A GEOMETRIC PERSPECTIVE

In this section we complement our discussion of (1) in Sec. 2 with a geometric interpretation, which allows us to make a connection between the decision version of (1) to the well known problem of completely positive matrix factorization [Berman and Shaked-Monderer, 2003, Dickinson, 2013, Dickinson and Gijben, 2014]. The decision version of the optimization problem in (1) is formulated as follows:

Definition 1 (SSymNMF). Given a matrix $P \in \mathbb{R}_+^{n \times n} \cap \mathbb{S}_+^n$, can it be factorized as $P = WW^T$ for some integer k such that $W \in \mathbb{R}_+^{n \times k}$ and $W\mathbf{1}_k = \mathbf{1}_n$?

Clearly, for $P \in \mathbb{S}_+^n$, we can decompose it as $P = UU^T$, where $U \in \mathbb{R}^{n \times r}$, $r = \text{rank}(P)$. Let $\mathbf{u}_i$ be the ith row vector of U; then P is the *Gram matrix* of a set of r dimensional vectors $\mathcal{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_n\} \subseteq \mathbb{R}^r$. Similarly, WW^T can be understood as the Gram matrix of a set of k dimensional vectors $\mathcal{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_n\} \subseteq \mathbb{R}^k$, where $\mathbf{w}_i$ is the ith row vector of W. The simplex constraint in (1) further restricts each $\mathbf{w}_i \in \Delta^{k-1}$, i.e., $\mathbf{w}_i$ resides in the $k-1$ dimensional probability simplex. Hence equivalently, SSymNMF asks the following question:

Definition 2. *Given a set of n instances $\mathcal{U} = \{\mathbf{u}_i\}_{i=1}^n \subseteq \mathbb{R}^r$, does there exist an integer k and an embedding $\mathcal{T}$: $\mathbb{R}^r \to \Delta^{k-1}$, such that inner product is preserved under $\mathcal{T}$, i.e., $\forall i, j \in [n], \langle \mathbf{u}_i, \mathbf{u}_j \rangle = \langle \mathcal{T}(\mathbf{u}_i), \mathcal{T}(\mathbf{u}_j) \rangle$?*

An affirmative answer to SSymNMF will give a certificate of the existence of such embedding $\mathcal{T}$: $\mathcal{T}(\mathbf{u}_i) = \mathbf{w}_i, \forall i \in [n]$. The goal of (1) can thus be understood as follows: find an embedding into the probability simplex such that the discrepancy of inner products between the image space and the original space is minimized. The fact that inner product is preserved immediately implies that distances between every pair of instances are also preserved. If $k = r$, such an embedding is also known as an *isometry*. In this case $\mathcal{T}$ is unitary. Note in the above definition of SSymNMF we do not restrict that $k = r$, and $\mathcal{T}$ does not have to be linear.

SSymNMF is closely connected to the strong membership problem for completely positive matrices [Berman and Plemmons, 1994, Berman and Shaked-Monderer, 2003]. A completely positive matrix is a matrix P that can be factorized as $P = WW^T$ where $W \in \mathbb{R}_+^{n \times k}$ for some k. The set of completely positive matrices forms a convex cone, and is known as the completely positive cone (CP cone). From this definition we can see that the decision version of SymNMF corresponds to the strong membership problem of the CP cone, which has recently been shown by Dickinson and Gijben [2014] to be NP-hard. Geometrically, the strong membership problem for CP matrices asks the following question:

Definition 3 (SMEMCP). *Given a set of n instances $\mathcal{U} = \{\mathbf{u}_i\}_{i=1}^n \subseteq \mathbb{R}^r$, does there exist an integer k and an embedding $\mathcal{T}$: $\mathbb{R}^r \to \mathbb{R}_+^k$, such that inner product is preserved under $\mathcal{T}$, i.e., $\forall i, j \in [n], \langle \mathbf{u}_i, \mathbf{u}_j \rangle = \langle \mathcal{T}(\mathbf{u}_i), \mathcal{T}(\mathbf{u}_j) \rangle$?*

Note that the only difference between SSymNMF and SMEMCP lies in the range of the image space: the former asks for an embedding in Δ^{k-1} while the latter only asks for embedding to reside in $\mathbb{R}_+^k$. SSymNMF is therefore conjectured to be NP-hard as well, but this assertion has not been formally proved yet. A simple reduction from SMEMCP to SSymNMF does not work: a "yes" answer to the former does not imply a "yes" answer to the latter.

3.2 ALGORITHM

We list the pseudocode of the FW method in Alg. 1 and discuss how Alg. 1 can be efficiently applied and implemented to solve SSymNMF. We start by deriving the gradient of $f(W) = \frac{1}{4}||P - WW^T||_F^2$:

$$\nabla f(W) = (WW^T - P)W \in \mathbb{R}^{n \times k} \qquad (7)$$

The normalization constraint keeps the feasible set decomposable (even though the objective function f is not decomposable): it can be equivalently represented as the product of n probability simplices $\Delta^{k-1} \times \cdots \times \Delta^{k-1} =: \Pi_n \Delta^{k-1}$. Hence at the t-th iteration of the algorithm we solve the following linear optimization problem over $\Pi_n \Delta^{k-1}$:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \text{tr}\left(S^T \nabla f(W^{(t)})\right) \\ \text{subject to} \quad & S \in \Pi_n \Delta^{k-1} \end{aligned} \qquad (8)$$

Because of the special structure of the constraint set in (8), given $\nabla f(W)$, we can efficiently compute the two key quantities $\mathbf{x}^{(t+1)}$ and g_t in Alg. 1 in $O(nk)$ time and $O(1)$ space. The pseudocode is listed in Alg. 2. The key observation that allows us to achieve this efficient implementation is that at each iteration $t \in [T]$, $\text{LMO}(\mathbf{x}^{(t)})$ is guaranteed to be a sparse matrix that contains exactly one 1 in each row. The time complexity of Alg. 2 is $3nk$, which can be further reduced to $2nk$ by additional $O(n)$ space to store the index of the nonzero element at each row of $\text{LMO}(\mathbf{x}^{(t)})$. Alg. 1, together with Alg. 2 as a subprocedure, is very efficient while at the same time being simple to implement. Furthermore, it does not have any hyperparameter to be tuned: this is in sharp contrast with the penalty method.

Algorithm 1 Frank-Wolfe algorithm (non-convex variant)

Input: Initial point $\mathbf{x}^{(0)} \in \mathcal{D}$, approximation parameter $\varepsilon > 0$

1: **for** $t = 0$ to T **do**
2:     Compute $\mathbf{s}^{(t)} = \text{LMO}(\mathbf{x}^{(t)}) := \arg\min_{\mathbf{s} \in \mathcal{D}} \mathbf{s}^T \nabla f(\mathbf{x}^{(t)})$
3:     Compute update direction $\mathbf{d}_t := \mathbf{s}^{(t)} - \mathbf{x}^{(t)}$
4:     Compute FW-gap $g_t := -\nabla f(\mathbf{x}^{(t)})^T \mathbf{d}_t$
5:     **if** $g_t \le \varepsilon$ **then return** $\mathbf{x}^{(t)}$
6:     Compute $\gamma_t := \min\{g_t/C, 1\}$ for any $C \ge \bar{C}_f$ (defined in (10))
7:     Update $\mathbf{x}^{(t+1)} := \mathbf{x}^{(t)} + \gamma_t \mathbf{d}_t$
8: **end for**
9: **return** $\mathbf{x}^{(T)}$

Algorithm 2 Compute next iterate and the gap function

Input: $\nabla f(\mathbf{x}^{(t)}), \mathbf{x}^{(t)}$
1: $g_t := \langle \nabla f(\mathbf{x}^{(t)}), \mathbf{x}^{(t)} \rangle$
2: **for** $i = 1$ to n **do**
3:     $g_t \leftarrow g_t - \min_{j \in [k]} \nabla f(\mathbf{x}^{(t)})_{ij}$
4: **end for**
5: Compute $\gamma_t := \min\{g_t/C, 1\}$
6: $\mathbf{x}^{(t+1)} := (1 - \gamma_t)\mathbf{x}^{(t)}$
7: **for** $i = 1$ to n **do**
8:     $j_i := \arg\min_{j \in [k]} \nabla f(\mathbf{x}^{(t)})_{ij}$
9:     $\mathbf{x}^{(t+1)}_{ij_i} \leftarrow \mathbf{x}^{(t+1)}_{ij_i} + \gamma_t$
10: **end for**
11: **return** $\mathbf{x}^{(t+1)}, g_t$

3.3 CONVERGENCE ANALYSIS

In this section we provide a non-asymptotic convergence rate for the FW algorithm for solving (1) and derive a tight bound for its curvature constant. Our analysis is based on the recent work [Lacoste-Julien, 2016], where we redefine the curvature constant so that the new definition works in both convex and nonconvex settings. Due to space limit, we only provide partial proofs for theorems and lemmas derived in this paper, and refer readers to supplementary material for all detailed proofs.

When applied to smooth convex constrained optimization problems, the FW algorithm is known to converge to an ε-approximate solution in $O(1/\varepsilon)$ iterations. However the convergence of global optimality is usually unrealistic to hope for in nonconvex optimization, where even checking local optimality itself can be computationally intractable [Murty and Kabadi, 1987]. Clearly, to talk about convergence, we need to first establish a convergence criterion. For unconstrained nonconvex problems, the norm of the gradient $||\nabla f||$ has been used to measure convergence [Ghadimi et al., 2016, Ghadimi and Lan, 2016], since $\lim_{t \to \infty} ||\nabla f(\mathbf{x}^{(t)})|| = 0$ means every limit point of the sequence $\{\mathbf{x}^{(t)}\}$ is a stationary point. But such a convergence criterion is not appropriate for constrained problems because a stationary point can lie on the boundary of the feasible region while not having a zero gradient. To address this issue, we will use the FW-gap $g(\mathbf{x})$ as a measure of convergence. Note that the gap $g(\mathbf{x})$ works as an optimality gap only if the original problem is convex. To see why this is also a good measure of convergence for constrained nonconvex problems, we first prove the following theorem:

Theorem 1. Let f be a differentiable function and $\mathcal{D}$ be a convex compact domain. Define $g(\mathbf{x}) := \max_{\hat{\mathbf{s}} \in \mathcal{D}} \nabla f(\mathbf{x})^T(\mathbf{x} - \hat{\mathbf{s}})$. Then $\forall \mathbf{x} \in \mathcal{D}, g(\mathbf{x}) \geq 0$ and $g(\mathbf{x}) = 0$ iff $\mathbf{x}$ is a Karush–Kuhn–Tucker (KKT) point.

Proof. To see $g(\mathbf{x}) \geq 0$, we have:

$$g(\mathbf{x}) := \max_{\hat{\mathbf{s}} \in \mathcal{D}} \nabla f(\mathbf{x})^T(\mathbf{x} - \hat{\mathbf{s}}) \geq \nabla f(\mathbf{x})^T(\mathbf{x} - \mathbf{x}) = 0$$

Reformulate the original constrained problem in the following way:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbb{I}_{\mathcal{D}}(\mathbf{x}) \leq 0 \end{aligned} \quad (9)$$

where $\mathbb{I}_{\mathcal{D}}(\mathbf{x})$ is the indicator function of $\mathcal{D}$ which takes value 0 iff $\mathbf{x} \in \mathcal{D}$ otherwise ∞. Define $\mathcal{N}_{\mathcal{D}}(\mathbf{x})$ as the *normal cone* at $\mathbf{x}$ in a convex set $\mathcal{D}$:

$$\mathcal{N}_{\mathcal{D}}(\mathbf{x}) := \{\mathbf{z} \mid \mathbf{z}^T \mathbf{x} \geq \mathbf{z}^T \mathbf{y}, \forall \mathbf{y} \in \mathcal{D}\}$$

and realize the fact that the subdifferential of the indicator function when $\mathbf{x} \in \mathcal{D}$ is precisely the normal cone at $\mathbf{x}$, i.e., $\partial \mathbb{I}_{\mathcal{D}}(\mathbf{x}) = \mathcal{N}_{\mathcal{D}}(\mathbf{x})$, we have:

$$g(\mathbf{x}) = 0 \iff \max_{\mathbf{y} \in \mathcal{D}} \nabla f(\mathbf{x})^T(\mathbf{x} - \mathbf{y}) = 0$$

$$\iff \forall \mathbf{y} \in \mathcal{D}, \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) \geq 0 \iff -\nabla f(\mathbf{x}) \in \mathcal{N}_{\mathcal{D}}(\mathbf{x})$$

Note that the normal cone for a convex set is a convex cone, which implies that $\exists \lambda \geq 0$, s.t.,

$$\nabla f(\mathbf{x}) + \lambda \partial \mathbb{I}_{\mathcal{D}}(\mathbf{x}) = 0 \iff \nabla_{\mathbf{x}} \mathcal{L}(x, \lambda) = 0$$

where $\mathcal{L}(\mathbf{x}, \lambda) := f(\mathbf{x}) + \lambda \mathbb{I}_{\mathcal{D}}(\mathbf{x})$ is the Lagrangian of (9). By construction, $\lambda \geq 0$ satisfies the dual feasibility condition and $\mathbf{x} \in \mathcal{D}$ satisfies the primal feasibility condition, which also means the complementary slackness is satisfied, i.e., $\lambda \mathbb{I}_{\mathcal{D}}(\mathbf{x}) = 0$. It follows that $g(\mathbf{x}) = 0$ iff $\mathbf{x}$ is a KKT point of (9). ■

Remark. Note that in Thm. 1 we do not assume f to be convex. In fact we can also relax the differentiability of f, as long as the subgradient exists at every point of f. The proof relies on the fact that $g(\mathbf{x}) = 0$ implies $-\nabla f(\mathbf{x})$ is in the normal cone at $\mathbf{x}$. Thm. 1 justifies the use of $g(\mathbf{x})$ as a convergence measure: if $\lim_{t \to \infty} g(\mathbf{x}^{(t)}) = 0$, then by the continuity of g, every limit point of $\{\mathbf{x}^{(k)}\}$ is a KKT point of f. Since being a KKT point is also a sufficient condition for optimality when f is convex, Thm. 1 also recovers the case where $g(\mathbf{x})$ is used as convergence measure for convex problems.

For a continuously differentiable function f, we now extend the definition of curvature constant as follows:

$$\bar{C}_f := \sup_{\substack{\mathbf{x}, \mathbf{s} \in \mathcal{D}, \\ \gamma \in (0,1], \\ \mathbf{y} = \mathbf{x} + \gamma(\mathbf{s} - \mathbf{x})}} \frac{2}{\gamma^2} \left| f(\mathbf{y}) - f(\mathbf{x}) - \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) \right|$$

$$(10)$$

Clearly $\bar{C}_f \geq 0$ and the new definition reduces to C_f when f is a convex function. In general we have $\bar{C}_f \geq$

C_f for any function f. Again, $\bar{C}_f$ measures the relative deviation of f from its linear approximation, and is still affine invariant. The difference of $\bar{C}_f$ from the original C_f becomes clear when f is concave: in this case $C_f = 0$, but $\bar{C}_f > 0$ and $\bar{C}_f = C_{-f}$. On the downside, the fact that $\bar{C}_f \geq C_f$ means our asymptotic bound is of constant times larger than the original one proved by Lacoste-Julien [2016], but they share the same asymptotic rate in terms of the given precision parameter ε. Finally, $\bar{C}_f = 0$ iff f is affine. As in [Jaggi, 2013, Lacoste-Julien et al., 2013], for a smooth function f with Lipschitz constant L over compact set $\mathcal{D}$, we can bound $\bar{C}_f$ in terms of L:

Lemma 1. Let f be a L-smooth function over a convex compact domain $\mathcal{D}$, and define $\operatorname{diam}(\mathcal{D}) := \sup_{\mathbf{x}, \mathbf{y} \in \mathcal{D}} ||\mathbf{x} - \mathbf{y}||$. Then $\bar{C}_f \leq \operatorname{diam}^2(\mathcal{D}) L$.

The proof of Lemma 1 does not require f to be convex. Furthermore, f does not need to be second-order differentiable — being smooth is sufficient. We proceed to derive a convergence bound for Alg. 1 using our new $\bar{C}_f$, which better reflects the geometric nonlinearity of nonconvex functions. The main idea of the proof is to bound the decrease of the gap function by minimizing a quadratic function iteratively.

Theorem 2. Consider the problem (2) where f is a continuously differentiable function that is potentially nonconvex, but has a finite curvature constant $\bar{C}_f$ as defined by (10) over the compact convex domain $\mathcal{D}$. Consider running Frank-Wolfe (Algo. 1), then the minimal FW gap $\tilde{g}_T := \min_{0 \leq t \leq T} g_t$ encountered by the iterates during the algorithm after T iterations satisfies:

$$\tilde{g}_T \leq \frac{\max\{2h_0 \bar{C}_f, \sqrt{2h_0 \bar{C}_f}\}}{\sqrt{T+1}}, \quad \forall T \geq 0 \qquad (11)$$

where $h_0 := f(\mathbf{x}^{(0)}) - \min_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x})$ is the initial global suboptimality. It thus takes at most $O(1/\varepsilon^2)$ iterations to find an approximate KKT point with gap smaller than ε.

We comment that the $O(1/\varepsilon^2)$ convergence rate is the same as the one provided by Lacoste-Julien [2016] using the original curvature constant C_f for smooth nonconvex functions, and it is also analogous to the ones derived for gradient descent for unconstrained smooth problems and (accelerated) projected gradient descent for constrained smooth problems. The convergence rate of Alg. 1 depends on the curvature constant of f over $\mathcal{D}$. We now bound the smoothness constant L and the diameter of the feasible set in (1).

Bound on smoothness constant. The objective function in (1) is second-order differentiable, hence we can bound the smoothness constant by bounding the spectral norm of the Hessian instead:

Lemma 2 (Nesterov [2013]). Let f be twice differentiable. If $||\nabla^2 f(\mathbf{x})||_2 \leq L$ for all $\mathbf{x}$ in the domain, then f is L-smooth.

Note that (7) is a matrix function of a matrix variable, whose Hessian is a matrix of order $nk \times nk$. Although one can compute all the elements of the Hessian by computing all the partial derivatives $\partial \nabla f(W)_{ij} / \partial W_{st}$ separately, this approach is tedious and may hide the structure of the Hessian matrix. Instead we apply matrix differential calculus [Magnus and Neudecker, 1985, Magnus, 2010] to derive the Hessian:

Lemma 3. Let $f(W) = \frac{1}{4}||P - WW^T||_F^2$ and define $\nabla^2 f(W) := \partial \operatorname{vec} \nabla f(W) / \partial \operatorname{vec} W$. Then:

$$\nabla^2 f(W) = W^T W \otimes I_n + I_k \otimes (WW^T - P)$$
$$+ (W^T \otimes W) K_{nk} \qquad (12)$$

where K_{nk} is a commutation matrix such that $K_{nk} \operatorname{vec} W = \operatorname{vec} W^T$.

The derivation of the above lemma uses the matrix differential calculus with basic properties of tensors and commutation matrices. Before we proceed, we first present a lemma that will be useful to bound the spectral norm of the above Hessian matrix:

Lemma 4. $\sup_{\substack{W \geq 0, \\ W\mathbf{1}_k = \mathbf{1}_n}} ||W^T W||_2 = n$.

We are now ready to bound the spectral norm of the Hessian $\nabla^2 f(W)$ and use it to bound the smoothness constant of f.

Lemma 5. Let $c := ||P||_2$. $f = \frac{1}{4}||P - WW^T||_F^2$ is $(3n + c)$-smooth on $\mathcal{D} = \{W \in \mathbb{R}_+^{n \times k} \mid W\mathbf{1}_k = \mathbf{1}_n\}$.

The above lemma follows from Lemma 2 where we use Lemma 4 to help bound the spectral norm of the Hessian matrix in (12).

Bound on diameter of $\mathcal{D}$. The following lemma can be easily shown:

Lemma 6. Let $\mathcal{D} = \{W \in \mathbb{R}_+^{n \times k} \mid W\mathbf{1}_k = \mathbf{1}_n\}$. Then $\operatorname{diam}^2(\mathcal{D}) = 2n$ with respect to the Frobenius norm.

Combining Lemma 6 with Lemma 5 and assuming c is a constant that does not depend on n, we immediately have $\bar{C}_f \leq 2n(3n + c) = O(n^2)$ by Lemma 1. A natural question to ask is: can we get better dependency on n in the upper bound for $\bar{C}_f$ given the special structure that $\mathcal{D} = \Pi_n \Delta^{k-1}$? The answer is negative, as we can prove the following lower bound on the Hessian:

Lemma 7. $\inf_{\substack{W \geq 0, \\ W\mathbf{1}_k = \mathbf{1}_n}} ||\nabla^2 f(W)||_2 \geq n/k^2 - c$.

Using Lemma 7, we can prove a tight bound on our curvature constant $\bar{C}_f$:

Theorem 3. The curvature constant $\bar{C}_f$ for $f = \frac{1}{4}\|P - WW^T\|_F^2$ on $\mathcal{D} = \{W \in \mathbb{R}_+^{n \times k} \mid W\mathbf{1}_k = \mathbf{1}_n\}$ satisfies:

$$2n(n/k^2 - c) \leq \bar{C}_f \leq 2n(3n + c)$$

where $c := \|P\|_2$. Specifically, we have $\bar{C}_f = \Theta(n^2)$.

Proof. The upper bound part is clear by combining Lemma 1 and Lemma 5. We only need to show the lower bound. Since f is twice-differentiable, we have:

$$
\begin{aligned}
\bar{C}_f &:= \sup_{\substack{\mathbf{x},\mathbf{s}\in\mathcal{D}, \\ \gamma\in(0,1], \\ \mathbf{y}=\mathbf{x}+\gamma(\mathbf{s}-\mathbf{x})}} \frac{2}{\gamma^2}\left|f(\mathbf{y}) - f(\mathbf{x}) - \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x})\right| \\
&= \sup_{\substack{\mathbf{x},\mathbf{s}\in\mathcal{D}, \\ \gamma\in(0,1], \\ \mathbf{y}=\mathbf{x}+\gamma(\mathbf{s}-\mathbf{x})}} \frac{2}{\gamma^2}\frac{1}{2}|(\mathbf{y}-\mathbf{x})^T\nabla^2 f(\xi)(\mathbf{y}-\mathbf{x})| \\
&\geq \sup_{\substack{\mathbf{x},\mathbf{s}\in\mathcal{D}, \\ \gamma\in(0,1], \\ \mathbf{y}=\mathbf{x}+\gamma(\mathbf{s}-\mathbf{x})}} \frac{1}{\gamma^2}\|\mathbf{y}-\mathbf{x}\|_2^2 \cdot \inf_{\xi\in\mathcal{D}}\|\nabla^2 f(\xi)\|_2 \\
&\geq \frac{1}{\gamma^2}\gamma^2 \mathrm{diam}^2(\mathcal{D})\left(\frac{n}{k^2} - c\right) = 2n\left(\frac{n}{k^2} - c\right)
\end{aligned}
$$

The second equality is due to the mean-value theorem, and the third inequality holds by the definition of $\inf$. The last inequality follows from Lemma 6 and 7. $\blacksquare$

Combining all the analysis above and using Alg. 2 to implement the linear minimization oracle in Alg. 1, we can bound the time complexity of the FW algorithm to solve (1):

Corollary 1. The FW algorithm (Alg. 1) achieves an ε-approximate KKT point of (1) in $O(n^3 k/\varepsilon^2)$ time.

Proof. In each iteration Alg. 1 takes $O(nk)$ time to compute the gap function as well as the next iterate. Based on Thm. 2, the iteration complexity to achieve an ε-approximate KKT point is $O(\bar{C}_f/\varepsilon^2)$. The result follows from Thm. 3 showing that $\bar{C}_f = \Theta(n^2)$. $\blacksquare$

4 A FAILURE CASE OF Frank-Wolfe ON A NONSMOOTH CONVEX PROBLEM

For convex problems, the theoretical convergence guarantee of Frank-Wolfe algorithm and its variants depends crucially on the smoothness of the objective function: [Freund and Grigas, 2016, Garber and Hazan, 2015, Harchaoui et al., 2015, Nesterov et al., 2015] require the gradient of the objective function to be Lipschitz continuous or Hölder continuous; the analysis in [Jaggi, 2013] requires a finite curvature constant C_f. The Lipschitz

continuous gradient condition is sufficient for the convergence analysis, but not necessary: Odor et al. [2016] shows that Frank-Wolfe works for a Poisson phase retrieval problem, where the gradient of the objective is not Lipschitz continuous. As we discuss in the last section, Lipschitz continuous gradient implies a finite curvature constant. Hence an interesting question to ask is: does there exist a constrained convex problem with unbounded curvature constant such that the Frank-Wolfe algorithm can find its optimal solution? Surprisingly, the answer to the above question is affirmative. But before we give the example, we first construct an example where the FW algorithm fails when the objective function is convex but nonsmooth.

We first construct a very simple example where the objective function is a piecewise linear and the constraint set is a polytope. We will show that when applied to this simple problem, the FW algorithm, along with its line search variant and its fully corrective variant, do not even converge to the optimal solution. In fact, as we will see shortly, the limit point of the sequence can be arbitrarily far away from the global optimum. Consider the following convex, constrained optimization problem:

$$\underset{x_1,x_2}{\text{minimize}} \quad \max\{5x_1 + x_2, -5x_1 + x_2\} \tag{13}$$

$$\text{subject to} \quad x_2 \leq 3, 3x_1 + x_2 \geq 0, -3x_1 + x_2 \geq 0$$

We plot the objective function of this example in the left figure of Fig. 1. The unique global optimum for this problem is given by $\mathbf{x}^* = (0,0)$ with $f(\mathbf{x}^*) = 0$. The feasible set $\mathcal{D}$ contains three vertices: $(-1,3)$, $(1,3)$ and $(0,0)$. If we apply the FW algorithm to this problem, it is straightforward to verify that the LMO($\mathbf{x}$) is given by:

$$\text{LMO}(\mathbf{x}) = \begin{cases} (-1,3) & x_1 > 0 \\ (1,3) & x_1 < 0 \\ \{(-1,3),(1,3),(0,0)\} & x_1 = 0 \end{cases}$$

Note that when $x_1 = 0$, the function is not differentiable, and the subdifferential is given by $\mathrm{conv}\{(5,1),(-5,1)\}$. In this case the FW algorithm chooses arbitrary subgradient from the subdifferential and computes the corresponding LMO. From the fundamental theorem of linear programming, it is easy to see that when $x_1 = 0$, LMO($\mathbf{x}$) can be any of the three corners depending on the choice of subgradient at $\mathbf{x}$. Now suppose the FW algorithm stops in T iterations. For any initial point $\mathbf{x}^{(0)} = (x_1^{(0)}, x_2^{(0)})$ where $x_1^{(0)} \neq 0$, the final point output by the FW algorithm will be a convex combination of $(-1,3)$, $(1,3)$, $(0,0)$ and $\mathbf{x}^{(0)}$. Let $\{\gamma_t\}_{t=1}^T$ be the sequence of step sizes chosen by the FW algorithm. Then we can easily check that $x_2^{(T)} \geq 3 - \prod_{t=1}^T (1 - \gamma_t)(1 - x_2^{(0)}/3) \to 3$ as
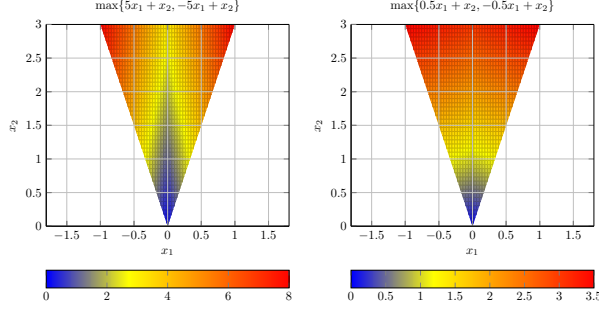
Figure 1: Two convex examples of piecewise linear objective functions on a convex polytope. The FW algorithm fails on the left example but works on the right one. The objective functions of both examples have unbounded curvature constants.

$T \to \infty$. Note that we can readily change this example by extending $\mathcal{D}$ so that the distance between $\mathbf{x}^{(T)}$ and the optimum $\mathbf{x}^*$ becomes arbitrarily large. Furthermore, both the line search and the fully-corrective variants fail since the vertices picked by the algorithm remains the same: $\{(-1, 3), (1, 3)\}$. Finally, for any regular probability distribution that is absolutely continuous w.r.t. the Lebesgue measure, with probability 1 the initial points sampled from the distribution will converge to suboptimal solutions. As a comparison, it can be shown that the subgradient method works for this problem since the function f itself is Lipschitz continuous [Nesterov, 2013].

Pick $\mathbf{x} = (-\varepsilon, \delta), \mathbf{s} - \mathbf{x} = (1, 0)$ and $\mathbf{y} = \mathbf{x} + \gamma(\mathbf{s} - \mathbf{x}) = (\gamma - \varepsilon, \delta)$, where $\gamma > \varepsilon$, and plug them in the definition of the curvature constant C_f. We have:

$$C_f \geq \lim_{\varepsilon \to 0, \gamma > \varepsilon} \frac{2}{\gamma^2} (f(\mathbf{y}) - f(\mathbf{x}) - \nabla f(\mathbf{x})^T (\mathbf{y} - \mathbf{x}))$$
$$= \lim_{\gamma \to 0_+} \frac{20}{\gamma} = \infty$$

i.e., the curvature constant of this piecewise linear function is unbounded. The problem for this failure case of FW lies in the fact that the curvature constant is infinity.

On the other hand, we can also show that FW works even when $C_f = \infty$ by slightly changing the objective function while keeping the constraint set:

$$\begin{aligned} \underset{x_1, x_2}{\text{minimize}} \quad & \max\{\frac{1}{2}x_1 + x_2, -\frac{1}{2}x_1 + x_2\} \quad (14) \\ \text{subject to} \quad & x_2 \leq 3, 3x_1 + x_2 \geq 0, -3x_1 + x_2 \geq 0 \end{aligned}$$

The objective function of the second example is shown in the right figure of Fig. 1. Still, the unique global optimum for the new problem is given by $\mathbf{x}^* = (0, 0)$ with $f(\mathbf{x}^*) = 0$, but now the LMO($\mathbf{x}$) is:

$$\text{LMO}(\mathbf{x}) = (0, 0), \quad \forall \mathbf{x} \in \mathcal{D}$$

It is not hard to see that FW converges to the global optimum, and the curvature constant $C_f = \infty$ as well for this new problem. Combining with example (14), we can see that $C_f < \infty$ is not a necessary condition for the success of FW algorithm on convex problems, either. Piecewise linear functions form a rich class of objectives that are frequently encountered in practice, while depending on the structure of the problem, the FW algorithm may or may not work for them. This thus raises an interesting problem: *can we develop a necessary condition for the success of the FW algorithm on convex problems?* Another interesting question is, *can we develop sufficient conditions for piecewise linear functions under which the FW algorithm converges to global optimum?*

5 NUMERICAL RESULTS

We evaluate the effectiveness of the FW algorithm (Alg. 1) in solving (1) by comparing it with the penalty method [Zhao et al., 2015] and the projected gradient descent method (PGD) on 4 datasets (Table 1). These datasets are standard for clustering analysis: two of them are used in [Zhao et al., 2015], and we add two more datasets with various sizes to make a more comprehensive evaluation. The instances in each dataset are associated with true class labels. For each data set, we use the number of classes as the true number of clusters.

Table 1: Statistics about datasets.

Dataset	# inst. (n)	# feats. (p)	# clusters (k)
blood	748	4	10
yeast	1,484	8	10
satimage	4,435	36	6
pendigits	10,992	16	100

Given a data matrix $X \in \mathbb{R}^{n \times p}$, we use a Gaussian kernel with fixed bandwidth 1.0 to construct the co-cluster matrix P as $P_{ij} = \exp(-||\mathbf{x}_i - \mathbf{x}_j||_2^2)$. For each dataset, we use its number of clusters as the rank of the decomposition, i.e., $W \in \mathbb{R}^{n \times k}$. We implement the penalty method based on [Zhao et al., 2015], where we set the maximum number of inner loops to be 50, and choose the step factor for the coefficients of the two penalty terms to be 2. In each iteration of PGD, we use backtracking line search to choose the step size. For all three algorithms, the stop conditions are specified as follows: if the difference of function values in two consecutive iterations is smaller than the fixed gap $\varepsilon = 10^{-3}$, or the number of (outer) iterations exceed 50, then the algorithms will stop. Also, for each dataset, all three algorithms share the same initial point.

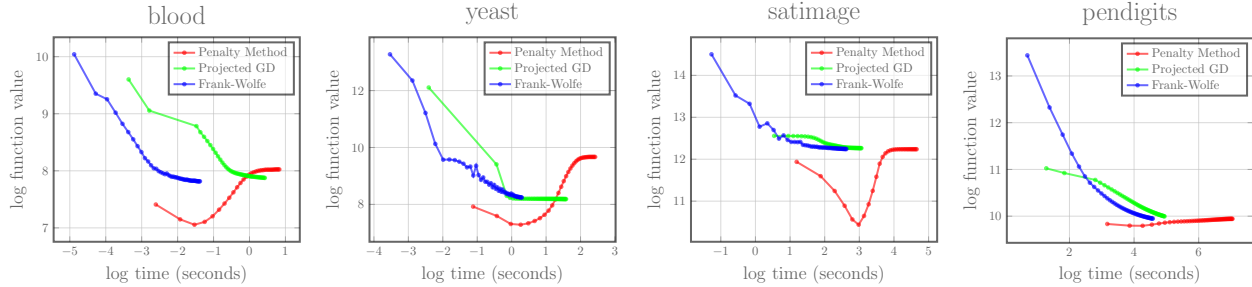We plot the convergence speed to local optimum of these

Figure 2: Convergence speed of three algorithms. The x-axis measures log-seconds and the y-axis measures log of objective function value. Note that the intermediate iterates of the penalty method are not feasible solutions, so we should only compare the convergent point of the penalty method with the other two.
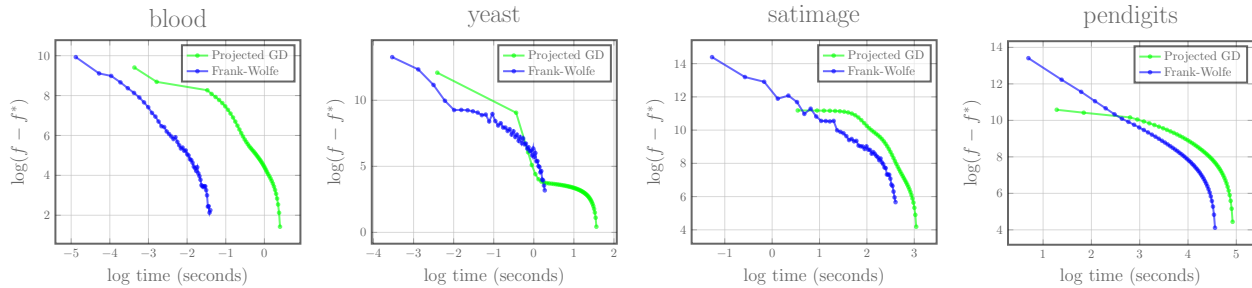

Figure 3: Convergence speed comparison between the projected gradient descent method and the FW algorithm. The x-axis measures log-seconds and the y-axis measures $\log(f - f^*)$, where f is the objective function value and f^* is the local optimum achieved by the algorithm. Note that the local optimum f^* achieved by PGD and FW can be different.

three algorithms in Fig. 2. Clearly, the penalty method is orders of magnitude slower than both PGD and the FW algorithm, and it usually converges to a worse solution. On the other hand, although the FW algorithm tends to have more iterations before it converges, due to its cheap computation in each iteration, it consistently takes less time than PGD. Another distinction between PGD and FW is that PGD, when implemented with backtracking line search, is a monotone descent method, while FW is not. For a better visualization to compare between the PGD and the FW algorithms, we omit the penalty method in Fig. 3, and draw the log-gap plot of the four datasets. We can confirm from Fig. 3 that both PGD and the FW algorithm have roughly the same order of convergence speed for solving (1), which is consistent with the theoretical result proved in the previous section (Thm. 2). However, the FW algorithm often converges faster than the PGD method, in terms of the gap between the objective function value and local optimum.

6   CONCLUSION

We propose a FW algorithm to solve the SymNMF problem under a simplicial constraint. Compared with existing solutions, the proposed algorithm enjoys a non-

asymptotic convergence guarantee to KKT points, is simple to implement, contains no hyperparameter to be tuned, and is also demonstrated to be much more efficient in practice. Theoretically, we establish a close connection of this problem to the famous completely positive matrix factorization by providing an equivalent geometric description. We also derive a tight bound on the curvature constant of this problem. As a side contribution, we give a pair of nonsmooth convex examples where the FW algorithm converges or fails to converge to its optimum. This result raises an interesting question w.r.t. the necessary condition of the success of the FW algorithm.

Acknowledgements

References

A. Berman and R. J. Plemmons. *Nonnegative matrices in the mathematical sciences*. SIAM, 1994.

A. Berman and N. Shaked-Monderer. *Completely positive matrices*. World Scientific, 2003.

C. Cartis, N. I. Gould, and P. L. Toint. On the complexity of steepest descent, newton's and regularized newton's methods for nonconvex unconstrained optimization problems. *SIAM Journal on Optimization*, 20 (6):2833–2852, 2010.

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

M. Collins, A. Globerson, T. Koo, X. Carreras, and P. L. Bartlett. Exponentiated gradient algorithms for conditional random fields and max-margin markov networks. *Journal of Machine Learning Research*, 9(Aug):1775–1822, 2008.

P. J. Dickinson and L. Gijben. On the computational complexity of membership problems for the completely positive cone and its dual. *Computational optimization and applications*, 57(2):403–415, 2014.

P. J. C. Dickinson. *The copositive cone, the completely positive cone and their generalisations*. Citeseer, 2013.

J. C. Dunn and S. Harshbarger. Conditional gradient algorithms with open loop step size rules. *Journal of Mathematical Analysis and Applications*, 62(2):432–444, 1978.

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

R. M. Freund and P. Grigas. New analysis and results for the frank–wolfe method. *Mathematical Programming*, 155(1-2):199–230, 2016.

D. Garber and E. Hazan. Faster Rates for the Frank-Wolfe Method over Strongly-Convex Sets. In *ICML*, pages 541–549, 2015.

S. Ghadimi and G. Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156(1-2):59–99, 2016.

S. Ghadimi, G. Lan, and H. Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155 (1-2):267–305, 2016.

Z. Harchaoui, A. Juditsky, and A. Nemirovski. Conditional gradient algorithms for norm-regularized smooth convex optimization. *Mathematical Programming*, 152 (1-2):75–112, 2015.

Z. He, S. Xie, R. Zdunek, G. Zhou, and A. Cichocki. Symmetric nonnegative matrix factorization: Algorithms and applications to probabilistic clustering. *IEEE Transactions on Neural Networks*, 22(12):2117–2131, 2011.

K. Huang, N. D. Sidiropoulos, and A. Swami. Non-negative matrix factorization revisited: Uniqueness and

algorithm for symmetric decomposition. *IEEE Transactions on Signal Processing*, 62(1):211–224, 2014.

M. Jaggi. *Sparse convex optimization methods for machine learning*. PhD thesis, ETH Zürich, 2011.

M. Jaggi. Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization. In *ICML*, pages 427–435, 2013.

D. Kuang, C. Ding, and H. Park. Symmetric nonnegative matrix factorization for graph clustering. In *SDM*, pages 106–117. SIAM, 2012.

D. Kuang, S. Yun, and H. Park. SymNMF: nonnegative low-rank approximation of a similarity matrix for graph clustering. *Journal of Global Optimization*, 62(3):545–574, 2015.

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

S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher. Block-Coordinate Frank-Wolfe Optimization for Structural SVMs. In *ICML*, pages 53–61, 2013.

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

E. S. Levitin and B. T. Polyak. Constrained minimization methods. *USSR Computational mathematics and mathematical physics*, 6(5):1–50, 1966.

W. Liu, N. Zheng, and X. Lu. Non-negative matrix factorization for visual coding. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 3, pages III–293. IEEE, 2003.

J. R. Magnus. On the concept of matrix derivative. *Journal of Multivariate Analysis*, 101(9):2200–2206, 2010.

J. R. Magnus and H. Neudecker. Matrix differential calculus with applications to simple, hadamard, and kronecker products. *Journal of Mathematical Psychology*, 29(4):474–492, 1985.

K. G. Murty and S. N. Kabadi. Some np-complete problems in quadratic and nonlinear programming. *Mathematical programming*, 39(2):117–129, 1987.

Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Y. Nesterov et al. Complexity bounds for primal-dual methods minimizing the model of objective function. *Center for Operations Research and Econometrics, CORE Discussion Paper*, 2015.

D. K. Nguyen, K. Than, and T. B. Ho. Simplicial nonnegative matrix factorization. In *Computing and Communication Technologies, Research, Innovation, and Vision*

for the Future (RIVF), 2013 IEEE RIVF International Conference on, pages 47–52. IEEE, 2013.

G. Odor, Y.-H. Li, A. Yurtsever, Y.-P. Hsieh, Q. Tran-Dinh, M. El Halabi, and V. Cevher. Frank-Wolfe works for non-Lipschitz continuous gradient objectives: Scalable poisson phase retrieval. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 6230–6234. IEEE, 2016.

A. Vandaele, N. Gillis, Q. Lei, K. Zhong, and I. Dhillon. Efficient and non-convex coordinate descent for symmetric nonnegative matrix factorization. *IEEE Transactions on Signal Processing*, 64(21):5571–5584, 2016.

S. A. Vavasis. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, 20(3): 1364–1377, 2009.

W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273. ACM, 2003.

H. Zhao, P. Poupart, Y. Zhang, and M. Lysy. Sof: Soft-cluster matrix factorization for probabilistic clustering. In *AAAI*, pages 3188–3195, 2015.

A MISSING PROOFS

Lemma 1. Let f be a L-smooth function over a convex compact domain $\mathcal{D}$, and define $\text{diam}(\mathcal{D}) := \sup_{\mathbf{x},\mathbf{y}\in\mathcal{D}} ||\mathbf{x}-\mathbf{y}||$. Then $\bar{C}_f \leq \text{diam}^2(\mathcal{D})L$.

Proof. Let $\forall \mathbf{x}, \mathbf{s} \in \mathcal{D}, \gamma \in (0,1]$, and $\mathbf{y} = \mathbf{x} + \gamma(\mathbf{s} - \mathbf{x})$. The smoothness of f implies that f is continuously differentiable, hence we have:

$$
\begin{aligned}
&\left| f(\mathbf{y}) - f(\mathbf{x}) - \nabla f(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) \right| \\
&= \left| \int_0^1 \left(\nabla f(\mathbf{x} + t(\mathbf{y} - \mathbf{x})) - \nabla f(\mathbf{x}) \right)^T (\mathbf{y} - \mathbf{x}) \, dt \right| \quad &\text{(Mean-value theorem)} \\
&\leq \int_0^1 \left| \left(\nabla f(\mathbf{x} + t(\mathbf{y} - \mathbf{x})) - \nabla f(\mathbf{x}) \right)^T (\mathbf{y} - \mathbf{x}) \right| dt \quad &\text{(Triangle inequality)} \\
&\leq \int_0^1 ||\nabla f(\mathbf{x} + t(\mathbf{y} - \mathbf{x})) - \nabla f(\mathbf{x})|| \cdot ||\mathbf{y} - \mathbf{x}|| \, dt \quad &\text{(Cauchy-Schwarz inequality)} \\
&\leq \int_0^1 tL\gamma^2 ||\mathbf{s} - \mathbf{x}||^2 \, dt \leq \frac{L\gamma^2}{2} \text{diam}^2(\mathcal{D}) \quad &\text{(Smoothness assumption of } f\text{)}
\end{aligned}
$$

It immediately follows that

$$
\bar{C}_f \leq \frac{2}{\gamma^2} \frac{L\gamma^2}{2} \text{diam}^2(\mathcal{D}) = \text{diam}^2(\mathcal{D})L
$$

∎

Theorem 2. Consider the problem (2) where f is a continuously differentiable function that is potentially nonconvex, but has a finite curvature constant $\bar{C}_f$ as defined by (10) over the compact convex domain $\mathcal{D}$. Consider running Frank-Wolfe (Algo. 1), then the minimal FW gap $\tilde{g}_T := \min_{0 \leq t \leq T} g_t$ encountered by the iterates during the algorithm after T iterations satisfies:

$$
\tilde{g}_T \leq \frac{\max\{2h_0\bar{C}_f, \sqrt{2h_0\bar{C}_f}\}}{\sqrt{T+1}}, \quad \forall T \geq 0 \tag{11}
$$

where $h_0 := f(\mathbf{x}^{(0)}) - \min_{\mathbf{x}\in\mathcal{D}} f(\mathbf{x})$ is the initial global suboptimality. It thus takes at most $O(1/\varepsilon^2)$ iterations to find an approximate KKT point with gap smaller than ε.

Proof. Let $\mathbf{y} := \mathbf{x} + \gamma\mathbf{d}$, where $\mathbf{d} := \mathbf{s} - \mathbf{x}$ is the update direction found by the LMO in Alg. 1. Using the definition of $\bar{C}_f$, we have:

$$
\begin{aligned}
f(\mathbf{y}) &= f(\mathbf{y}) - f(\mathbf{x}) - \gamma\nabla f(\mathbf{x})^T\mathbf{d} + f(\mathbf{x}) + \gamma\nabla f(\mathbf{x})^T\mathbf{d} \\
&\leq f(\mathbf{x}) + \gamma\nabla f(\mathbf{x})^T\mathbf{d} + \left| f(\mathbf{y}) - f(\mathbf{x}) - \gamma\nabla f(\mathbf{x})^T\mathbf{d} \right| \\
&\leq f(\mathbf{x}) + \gamma\nabla f(\mathbf{x})^T\mathbf{d} + \frac{\gamma^2}{2}\bar{C}_f
\end{aligned}
$$

Now using the definition of the FW gap $g(\mathbf{x})$ and for $\forall C \geq \bar{C}_f$, we get:

$$
f(\mathbf{y}) \leq f(\mathbf{x}) - \gamma g(\mathbf{x}) + \frac{\gamma^2}{2}\bar{C}_f, \quad \forall \gamma \in (0,1] \tag{15}
$$

Depending on whether $C > 0$ or $C = 0$, the R.H.S. of (15) is a either a quadratic function with positive second order coefficient or an affine function. In the first case, the optimal γ^* that minimizes the R.H.S. is $\gamma^* = g(\mathbf{x})/C$. In the second case, $\gamma^* = 1$. Combining the constraint that $\gamma^* \leq 1$, we have $\gamma^* = \min\{1, g(\mathbf{x})/C\}$. Thus we obtain:

$$
f(\mathbf{y}) \leq f(\mathbf{x}) - \min\left\{ \frac{g^2(\mathbf{x})}{2C}, \left(g(\mathbf{x}) - \frac{C}{2} \right) \mathbb{I}_{g(\mathbf{x})>C} \right\} \tag{16}
$$

(16) holds for each iteration in Alg. 1. A cascading sum of (16) through iteration step 1 to $T+1$ shows that:

$$f(\mathbf{x}^{(T+1)}) \leq f(\mathbf{x}^{(0)}) - \sum_{t=0}^{T} \min\left\{ \frac{g^2(\mathbf{x}^{(t)})}{2C}, \left(g(\mathbf{x}^{(t)}) - \frac{C}{2}\right) \mathbb{I}_{g(\mathbf{x}^{(t)})>C} \right\} \tag{17}$$

Define $\tilde{g}_T := \min_{0 \leq t \leq T} g(\mathbf{x}^{(t)})$ be the minimal FW gap in $T+1$ iterations. Then we can further bound inequality (17) as:

$$f(\mathbf{x}^{(T+1)}) \leq f(\mathbf{x}^{(0)}) - (T+1)\min\left\{ \frac{\tilde{g}_T^2}{2C}, \left(\tilde{g}_T - \frac{C}{2}\right) \mathbb{I}_{\tilde{g}_T>C} \right\} \tag{18}$$

We discuss two subcases depending on whether $\tilde{g}_T > C$ or not. The main idea is to get an upper bound on $\tilde{g}_T$ by showing that $\tilde{g}_T$ cannot be too large, otherwise the R.H.S. of (18) can be smaller than the global minimum of f, which is a contradiction. For the ease of notation, define $h_0 := f(\mathbf{x}^{(0)}) - \min_{\mathbf{x}\in\mathcal{D}} f(\mathbf{x})$, i.e., the initial gap to the global minimum of f.

Case I. If $\tilde{g}_T > C$ and $\tilde{g}_T - \frac{C}{2} \leq \frac{\tilde{g}_T^2}{2C}$, from (18), then:

$$0 \leq f(\mathbf{x}^{(T+1)}) - \min_{\mathbf{x}\in\mathcal{D}} f(\mathbf{x}) \leq f(\mathbf{x}^{(0)}) - \min_{\mathbf{x}\in\mathcal{D}} f(\mathbf{x}) - (T+1)(\tilde{g}_T - \frac{C}{2}) = h_0 - (T+1)(\tilde{g}_T - \frac{C}{2})$$

which implies

$$C < \tilde{g}_T \leq \frac{h_0}{T+1} + \frac{C}{2} \Rightarrow \tilde{g}_T \leq \frac{2h_0 C}{T+1} = O(1/T)$$

On the other hand, solving the following inequality:

$$C - \frac{C}{2} \leq \tilde{g}_T - \frac{C}{2} \leq \frac{\tilde{g}_T^2}{2C} \leq \frac{4h_0^2 C^2}{(T+1)^2}\frac{1}{2C}$$

we get

$$T+1 \leq 2h_0$$

This means that $\tilde{g}_T$ decreases in rate $O(1/T)$ only for at most the first $2h_0$ iterations.

Case II. If $\tilde{g}_T \leq C$ or $\tilde{g}_T - \frac{C}{2} > \frac{\tilde{g}_T^2}{2C}$. Similarly, from (18), we have:

$$0 \leq f(\mathbf{x}^{(T+1)}) - \min_{\mathbf{x}\in\mathcal{D}} f(\mathbf{x}) \leq f(\mathbf{x}^{(0)}) - \min_{\mathbf{x}\in\mathcal{D}} f(\mathbf{x}) - (T+1)\frac{\tilde{g}_T^2}{2C} = h_0 - (T+1)\frac{\tilde{g}_T^2}{2C}$$

which yields

$$\tilde{g}_T \leq \sqrt{\frac{2h_0 C}{T+1}}$$

Combining the two cases together, we get $\tilde{g}_T \leq \frac{2h_0 C}{T+1}$ if $T+1 \leq 2h_0$; otherwise $\tilde{g}_T \leq \sqrt{\frac{2h_0 C}{T+1}}$. Note that for $T \geq 0$, $\sqrt{T+1} \leq T+1$, thus we can further simplify the upper bound of $\tilde{g}_T$ as:

$$\tilde{g}_T \leq \frac{\max\{2h_0 C, \sqrt{2h_0 C}\}}{\sqrt{T+1}}$$

$$\blacksquare$$

Lemma 3. Let $f(W) = \frac{1}{4}||P - WW^T||_F^2$ and define $\nabla^2 f(W) := \partial \text{vec}\, \nabla f(W)/\partial \text{vec}\, W$. Then:

$$\nabla^2 f(W) = W^T W \otimes I_n + I_k \otimes (WW^T - P)$$
$$+ (W^T \otimes W)K_{nk} \tag{12}$$

where K_{nk} is a commutation matrix such that $K_{nk} \text{vec}\, W = \text{vec}\, W^T$.

Proof. Using the theory of matrix differential calculus, the Hessian of a matrix-valued matrix function is defined as:

$$\nabla^2 f(W) := \frac{\partial \operatorname{vec} \nabla f(W)}{\partial \operatorname{vec} W}$$

Using the differential notation, we can compute the differential of $\nabla f(W)$ as:

$$\mathrm{d}\nabla f(W) = \mathrm{d}(WW^T - P)W = (\mathrm{d}W)W^T W + W(\mathrm{d}W)^T W + WW^T \,\mathrm{d}W - P\,\mathrm{d}W$$

Vectorize both sides of the above equation and make use of the identity that $\operatorname{vec}(ABC) = (C^T \otimes A)\operatorname{vec} B$ for A, B, C with appropriate shapes, we get:

$$\operatorname{vec}\mathrm{d}\,\nabla f(W) = (W^T W \otimes I_n)\operatorname{vec}\mathrm{d}W + (W^T \otimes W)\operatorname{vec}\mathrm{d}W^T + (I_k \otimes (WW^T - P))\operatorname{vec}\mathrm{d}W$$

Let K_{nk} be a commutation matrix such that $K_{nk}\operatorname{vec} W = \operatorname{vec} W^T$. We can further simplify the above equation as:

$$\operatorname{vec}\mathrm{d}\nabla f(W) = \big(W^T W \otimes I_n + (W^T \otimes W)K_{nk} + I_k \otimes (WW^T - P)\big)\operatorname{vec}\mathrm{d}W \tag{19}$$

It then follows from the first identification theorem [Magnus and Neudecker, 1985, Thm. 6] that the Hessian is given by

$$\nabla^2 f(W) = \big(W^T W \otimes I_n + I_k \otimes (WW^T - P) + (W^T \otimes W)K_{nk}\big) \in \mathbb{R}^{nk \times nk}$$

As a sanity check, the first two terms in $\nabla^2 f(W)$ are clearly symmetric. The third term can be verified as symmetric as well by realizing that $K_{nk}^{-1} = K_{nk}^T$, and

$$W \otimes W^T = K_{nk}(W^T \otimes W)K_{nk}$$

∎

Lemma 4. $\sup_{\substack{W \geq 0, \\ W\mathbf{1}_k = \mathbf{1}_n}} \|W^T W\|_2 = n.$

Proof. $\forall W \geq 0$, if $W\mathbf{1}_k = \mathbf{1}_n$, then by the Courant-Fischer theorem:

$$
\begin{aligned}
\|W^T W\|_2 &:= \max_{\substack{\mathbf{v} \in \mathbb{R}^k, \\ \|\mathbf{v}\|_2 = 1}} \|W^T W \mathbf{v}\|_2 && \text{(Courant-Fischer theorem)} \\
&= \max_{\substack{\mathbf{v} \in \mathbb{R}^k_+, \\ \|\mathbf{v}\|_2 = 1}} \|W^T W \mathbf{v}\|_2 && \text{(Perron-Frobenius theorem)} \\
&\leq \max_{\substack{\mathbf{v} \in \mathbb{R}^k_+, \\ \|\mathbf{v}\|_\infty \leq 1}} \|W^T W \mathbf{v}\|_2 && (B_2(0,1) \subseteq B_\infty(0,1)) \\
&= \|W^T \mathbf{1}_n\|_2 && (W \geq 0, W\mathbf{1}_k = \mathbf{1}_n) \\
&\leq \|W^T \mathbf{1}_n\|_1 = n
\end{aligned}
$$

To achieve this upper bound, consider $W = \mathbf{1}_n e_1^T$, where e_1 is the first column vector of the identity matrix I_k. In this case $W^T W = e_1 \mathbf{1}_n^T \mathbf{1}_n e_1^T = n e_1 e_1^T$, which is a rank one matrix with a positive eigenvalue n. Hence $\sup \|W^T W\|_2 = n$. ∎

Lemma 5. Let $c := \|P\|_2$. $f = \frac{1}{4}\|P - WW^T\|_F^2$ is $(3n + c)$-smooth on $\mathcal{D} = \{W \in \mathbb{R}^{n \times k}_+ \mid W\mathbf{1}_k = \mathbf{1}_n\}$.

Proof. Recall that the spectral norm $\|\cdot\|_2$ is sub-multiplicative and the spectrum of $A \otimes B$ is the product of the spectrums of A and B. Using (12), we have:

$$
\begin{aligned}
\|\nabla^2 f(W)\|_2 &= \|W^T W \otimes I_n + I_k \otimes (WW^T - P) + (W^T \otimes W)K_{nk}\|_2 \\
&\leq \|W^T W \otimes I_n\|_2 + \|I_k \otimes (WW^T - P)\|_2 + \|(W^T \otimes W)K_{nk}\|_2 && \text{(Triangle inequality)} \\
&= \|W^T W\|_2\|I_n\|_2 + \|I_k\|_2\|WW^T - P\|_2 + \|W^T \otimes W\|_2\|K_{nk}\|_2 && \text{(submultiplicativity of } \|\cdot\|_2) \\
&= \|W^T W\|_2 + \|WW^T - P\|_2 + \|W^T \otimes W\|_2 && (\|I_n\|_2 = \|I_k\|_2 = \|K_{nk}\|_2 = 1) \\
&\leq 3\|W^T W\|_2 + \|P\|_2 && \text{(Triangle inequality)} \\
&\leq 3n + c && \text{(Lemma 4)}
\end{aligned}
$$

The result then follows from Lemma 2. ∎

Lemma 6. Let $\mathcal{D} = \{W \in \mathbb{R}_+^{n \times k} \mid W\mathbf{1}_k = \mathbf{1}_n\}$. Then $\text{diam}^2(\mathcal{D}) = 2n$ with respect to the Frobenius norm.

Proof.

$$\text{diam}^2(\mathcal{D}) = \sup_{W,Z \in \mathcal{D}} ||W - Z||_F^2$$

$$= \sup_{W,Z \in \mathcal{D}} \sum_{ij} (W_{ij} - Z_{ij})^2 = \sup_{W,Z \in \mathcal{D}} \sum_{ij} W_{ij}^2 + Z_{ij}^2 - 2W_{ij}Z_{ij}$$

$$\leq \sup_{\substack{W,Z \in \mathcal{D} \\ W,Z \in \mathcal{D}}} \sum W_{ij}^2 + Z_{ij}^2 \leq \sup_{\substack{W,Z \in \mathcal{D} \\ W,Z \in \mathcal{D}}} \sum W_{ij} + Z_{ij}$$

$$= 2n$$

Note that choosing $W = \mathbf{1}e_1^T$ and $Z = \mathbf{1}_n e_2^T$ make all the equalities hold in the above inequalities. Hence $\text{diam}^2(\mathcal{D}) = 2n$. ∎

Lemma 7. $\inf_{\substack{W \geq 0, \\ W\mathbf{1}_k = \mathbf{1}_n}} ||\nabla^2 f(W)||_2 \geq n/k^2 - c$.

Proof. For a matrix A, we will use $\sigma_i(A)$ to mean the ith largest singular value of A and $\lambda_{max}(A)$, $\lambda_{min}(A)$ to mean the largest and smallest eigenvalues of A, respectively. Recall $\nabla^2 f(W) = W^T W \otimes I_n + I_k \otimes (WW^T - P) + (W^T \otimes W)K_{nk}$. For $W \geq 0, W\mathbf{1}_k = \mathbf{1}_n$, let $r = \text{rank}(W)$. Clearly $r \geq 1$. We have the following inequalities hold:

$$||\nabla^2 f(W)||_2 = ||W^T W \otimes I_n + I_k \otimes (WW^T - P) + (W^T \otimes W)K_{nk}||_2$$

$$\geq \lambda_{max}\left(WW^T \otimes I_n + (W^T \otimes W)K_{nk}\right) + \lambda_{min}\left(I_k \otimes (WW^T - P)\right) \qquad \text{(Weyl's inequality)}$$

$$\geq \lambda_{max}(WW^T \otimes I_n) + \lambda_{min}\left((W^T \otimes W)K_{nk}\right) + \lambda_{min}\left(I_k \otimes (WW^T - P)\right)$$

$$= \lambda_{max}(WW^T) + \lambda_{min}(W^T \otimes W) + \lambda_{min}(WW^T - P)$$

$$\geq \lambda_{max}(WW^T) + \lambda_{min}(W^T \otimes W) + \lambda_{min}(WW^T) - \lambda_{max}(P)$$

$$= \sigma_1^2(W) + 2\sigma_r^2(W) - \lambda_{max}(P)$$

$$\geq \sigma_1^2(W) - c \qquad (||P||_2 \leq ||P||_F)$$

$$\geq \frac{1}{r}||W||_F^2 - c \qquad (r \cdot \sigma_1^2(W) \geq ||W||_F^2)$$

$$\geq \frac{1}{k}||W||_F^2 - c \qquad (\text{rank}(W) \leq k)$$

$$= \frac{1}{k}\sum_{i=1}^{n}\sum_{j=1}^{k} W_{ij}^2 - c$$

$$\geq \frac{1}{k}\sum_{i=1}^{n} k\left(\frac{\sum_{j=1}^{k} W_{ij}}{k}\right)^2 - c \qquad \text{(Cauchy ineq.)}$$

$$= \frac{n}{k^2} - c$$

where the first three inequalities all follow from Weyl's inequality. ∎