Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization

Martin Jaggi JAGGI@CMAP.POLYTECHNIQUE.FR
CMAP, École Polytechnique, Palaiseau, France

Abstract

We provide stronger and more general primal-dual convergence results for Frank-Wolfe-type algorithms (a.k.a. conditional gradient) for constrained convex optimization, enabled by a simple framework of duality gap certificates. Our analysis also holds if the linear subproblems are only solved approximately (as well as if the gradients are inexact), and is proven to be worst-case optimal in the sparsity of the obtained solutions.

On the application side, this allows us to unify a large variety of existing sparse greedy methods, in particular for optimization over convex hulls of an atomic set, even if those sets can only be approximated, including sparse (or structured sparse) vectors or matrices, low-rank matrices, permutation matrices, or max-norm bounded matrices.

We present a new general framework for convex optimization over *matrix factorizations*, where every Frank-Wolfe iteration will consist of a low-rank update, and discuss the broad application areas of this approach.

1. Introduction

Our work here addresses general constrained convex optimization problems of the form

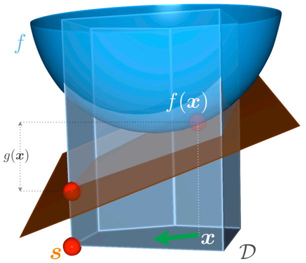
$$\min_{\boldsymbol{x}\in\mathcal{D}} f(\boldsymbol{x}) \ . \tag{1}$$

We assume that the objective function f is convex and continuously differentiable, and that the domain $\mathcal{D}$ is a compact convex subset of any vector space[1]. For such optimization problems, one of the simplest and earliest known iterative optimizers is given by the Frank-Wolfe method (1956), described in Algorithm 1, also known as the *conditional gradient method*.

Algorithm 1 Frank-Wolfe (1956)

```
Let $\boldsymbol{x}^{(0)} \in \mathcal{D}$
for $k = 0 \dots K$ do
    Compute $\boldsymbol{s} := \arg\min_{\boldsymbol{s}\in\mathcal{D}} \left\langle \boldsymbol{s}, \nabla f(\boldsymbol{x}^{(k)}) \right\rangle$
    Update $\boldsymbol{x}^{(k+1)} := (1-\gamma)\boldsymbol{x}^{(k)} + \gamma\boldsymbol{s}$,   for $\gamma := \frac{2}{k+2}$
end for
```

A step of this algorithm is illustrated in the inset figure: At a current position $\boldsymbol{x}$, the algorithm considers the linearization of the objective function, and moves towards a minimizer of this linear function (taken over the same domain).



In terms of convergence, it is known that the iterates of Algorithm 1 satisfy $f(\boldsymbol{x}^{(k)}) - f(\boldsymbol{x}^*) \leq O\big(\frac{1}{k}\big)$, for $\boldsymbol{x}^*$ being an optimal solution to (1) (Frank & Wolfe, 1956; Dunn & Harshbarger, 1978). In recent years, Frank-Wolfe-type methods have re-gained interest in several areas, fueled by the good scalability, and the crucial property that Algorithm 1 maintains its iterates as a convex combination of only few “atoms” $\boldsymbol{s}$, enabling e.g. sparse and low-rank solutions (since at most one new extreme point of the domain $\mathcal{D}$ is added in each step) see e.g. (Clarkson, 2010; Jaggi, 2011) for an overview.

Contributions. The contributions of this paper are two-fold: On the theoretical side, we give a convergence analysis for the general Frank-Wolfe algorithm guaranteeing small duality gap, and provide efficient certificates for the approximation quality (which are useful even for other optimizers). This result is obtained by extending the duality concept as well as the analysis of (Clarkson, 2010) to general Fenchel duality, and approximate linear subproblems. Furthermore, the presented analysis unifies several existing convergence results for different sparse greedy algorithm variants into one simplified proof. In contrast to existing convex optimization methods, our convergence analysis (as well as the algorithm itself) are fully *invariant* under any affine transformation/pre-conditioning

[1] Formally, we assume that the optimization domain $\mathcal{D}$ is a compact and convex subset of a Hilbert space $\mathcal{X}$, i.e. a Banach space equipped with an inner product $\langle .,. \rangle$.

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

of the input optimization problem (1).

On the practical side, we illustrate the broader applicability of Frank-Wolfe-type methods, when compared to their main competitors being projected gradient descent and proximal methods. Per iteration, Frank-Wolfe uses significantly less expensive *linear* subproblems compared to *quadratic* problems in the later, which can make the difference between simple and intractable for e.g. the dual of structural SVMs (Lacoste-Julien et al., 2013), or an order of magnitude iteration cost for the trace norm (leading eigenvector vs. SVD) (Jaggi & Sulovský, 2010).

We point out that all convex optimization problems over convex hulls of atomic sets (Chandrasekaran et al., 2012), which appear as the natural convex relaxations of combinatorial (NP-hard) "sparsity" problems, are directly suitable for Frank-Wolfe-type methods (using one atom per iteration), even when the domain can only be approximated. For optimization over vectors, prominent examples include optimizing over arbitrary norm-constrained domains (such as ℓ_1), as well as norms that induce *structured* sparsity of the approximate solutions, such as submodular polyhedra.

For matrix optimization problems, our presented approach results in simplified algorithms for optimizing over bounded matrix trace norm, arbitrary Schatten norms, or also permutation matrices and rotation matrices. Another particularly interesting application is convex optimization over bounded matrix max-norm, where no convergence guarantees were known previously. Finally, we present a new general framework for convex optimization over *matrix factorizations*, where every Frank-Wolfe iteration will consist of a low-rank update, and discuss applications for a broad range of such domains.

History and Related Work. The original Frank-Wolfe algorithm (1956) was introduced and analyzed for polyhedral domains $\mathcal{D}$ in $\mathbb{R}^n$ (given as an intersection of linear constraints, so that the subproblem becomes an LP). The original paper did not yet use the "fixed" step-size as in Algorithm 1, but instead relied on line-search on a quadratic upper bound on f. (Levitin & Polyak, 1966) coined the term *conditional gradient method* for the same algorithm, which (Demyanov & Rubinov, 1970) then generalized to arbitrary Banach spaces as in the setting here. Later (Dunn & Harshbarger, 1978) could show primal convergence with $\frac{1}{k}$ when only approximate linear minimizers of the subproblems are used, and (Patriksson, 1993) investigated several alternative variations of (non-)linear subproblems. Another variant using non-linear subproblems was proposed in (Zhang, 2003), in each iteration performing a line-search on f towards all "vertices" of the domain.

In the machine learning literature, algorithm variants for penalized (instead of constrained) problems were investigated by (Harchaoui et al., 2012; Zhang et al., 2012). For online optimization of non-smooth functions in the low-regret setting, a variant has recently been proposed by (Hazan & Kale, 2012), using randomized smoothing. (Tewari et al., 2011) and (Dudik et al., 2012, Appendix D) have recently studied Frank-Wolfe methods for atomic domains using similar ideas as in (Jaggi, 2011), but obtaining weaker convergence results. (Temlyakov, 2012) gives a recent comprehensive analysis of such greedy methods from the convex analysis perspective. To the best of our knowledge, none of the existing approaches could provide duality gap convergence guarantees, or affine invariance (except (Clarkson, 2010) for the simplex case). A block-coordinate generalisation of Frank-Wolfe has recently been proposed in (Lacoste-Julien et al., 2013).

2. The Duality Gap and Certificates

For any constrained convex optimization problem of the form (1), and a feasible point $\boldsymbol{x} \in \mathcal{D}$, we define the following simple surrogate duality gap

$$g(\boldsymbol{x}) := \max_{\boldsymbol{s}\in\mathcal{D}} \left\langle \boldsymbol{x} - \boldsymbol{s}, \nabla f(\boldsymbol{x}) \right\rangle . \qquad (2)$$

Convexity of f implies that the linearization $f(\boldsymbol{x}) + \left\langle \boldsymbol{s}-\boldsymbol{x}, \nabla f(\boldsymbol{x}) \right\rangle$ always lies below the graph of the function f, as again illustrated in Figure 1. This immediately gives the crucial property of the duality gap (2), as being a *certificate* for the current approximation quality, i.e. $g(\boldsymbol{x}) \geq f(\boldsymbol{x}) - f(\boldsymbol{x}^*)$.

While the value of an optimal solution $f(\boldsymbol{x}^*)$ is unknown in most problems of interest, the quantity $g(\boldsymbol{x})$ for any candidate $\boldsymbol{x}$ is often easy to compute. For example, the duality gap is "automatically" computed as a by-product of every iteration of the Frank-Wolfe Algorithm 1: Whenever $\boldsymbol{s}$ is a minimizer of the linearized problem at an arbitrary point $\boldsymbol{x}$, then this $\boldsymbol{s}$ is a certificate for the current duality gap $g(\boldsymbol{x}) = \left\langle \boldsymbol{x}-\boldsymbol{s}, \nabla f(\boldsymbol{x}) \right\rangle$.

Such certificates for the approximation quality are useful not only for the algorithms considered here, but in fact for any optimizer of a constrained problem of the form (1), e.g. as a stopping criterion, or to verify the numerical stability of an optimizer. This duality concept also extends to the more general case if f is convex but non-smooth. In this case, the gap is certified by a *subgradient* of f, see e.g. (Jaggi, 2011, Section 2.2).

Our defined duality gap (2) can also be interpreted as a special (and simplified) case of Fenchel duality. Using the Fenchel-Young (in)equality, the gap (2) can

Algorithm 2 Frank-Wolfe with Approximate Linear Subproblems, for Quality $\delta \geq 0$

```
   Let $\boldsymbol{x}^{(0)} \in \mathcal{D}$
   for $k = 0 \dots K$ do
      Let $\gamma := \frac{2}{k+2}$
      Find $\boldsymbol{s} \in \mathcal{D}$ s.t.
            $\left\langle \boldsymbol{s}, \nabla f(\boldsymbol{x}^{(k)}) \right\rangle \leq \min_{\hat{\boldsymbol{s}} \in \mathcal{D}} \left\langle \hat{\boldsymbol{s}}, \nabla f(\boldsymbol{x}^{(k)}) \right\rangle + \frac{1}{2}\delta\gamma C_f$
a)    (Optionally: Perform line-search for $\gamma$)
b)    Update $\boldsymbol{x}^{(k+1)} := (1-\gamma)\boldsymbol{x}^{(k)} + \gamma \boldsymbol{s}$
   end for
```

Algorithm 3 Line-Search for the Step-Size γ

```
 ... as Algorithm 2, except replacing line a) with
a')   $\gamma := \arg\min_{\gamma \in [0,1]} f\Big(\boldsymbol{x}^{(k)} + \gamma\big(\boldsymbol{s} - \boldsymbol{x}^{(k)}\big)\Big)$
```

Algorithm 4 Fully-Corrective Variant, Re-Optimizing over all Previous Directions *(with $\boldsymbol{s}^{(0)} := \boldsymbol{x}^{(0)}$)*

```
 ... as Algorithm 2, except replacing line b) with
b')   Update $\boldsymbol{x}^{(k+1)} := \arg\min_{\boldsymbol{x} \in \operatorname{conv}(\boldsymbol{s}^{(0)}, \dots, \boldsymbol{s}^{(k+1)})} f(\boldsymbol{x})$
```

be shown to be equal to the difference of $f(\boldsymbol{x})$ to the Fenchel conjugate function of f, if the corresponding dual variable is chosen to be the current (sub)gradient, see (Lacoste-Julien et al., 2013, Appendix D).

3. Frank-Wolfe Algorithms

Besides classical Frank-Wolfe (Algorithm 1), the following three algorithm variants are relevant. Later we will prove primal-dual convergence for all four algorithm variants together.

Approximating the Linear Subproblems. Depending on the domain $\mathcal{D}$, solving the linear subproblem $\min_{\boldsymbol{s} \in \mathcal{D}} \left\langle \boldsymbol{s}, \nabla f(\boldsymbol{x}^{(k)}) \right\rangle$ exactly can be too expensive. Algorithm 2 uses any approximate minimizer $\boldsymbol{s}$ instead, of an additive approximation quality at least $\varepsilon' := \frac{1}{2}\delta\gamma C_f = \frac{\delta C_f}{k+2}$ in step k. Here $\delta \geq 0$ is an arbitrary fixed accuracy parameter.

Line-Search for the Step-Size. Instead of using the pre-defined step-sizes $\gamma = \frac{2}{k+2}$, Algorithm 3 picks the best point on the line segment between the current iterate $\boldsymbol{x}^{(k)}$ and $\boldsymbol{s}$.

"Fully Corrective" Variant. Algorithm 4 depicts the harder-working variant of the Frank-Wolfe method, which after the addition of a new atom (or search direction) $\boldsymbol{s}$ re-optimizes the objective f over all previously used atoms. Here in step k, the current atom $\boldsymbol{s} = \boldsymbol{s}^{(k+1)}$ is still allowed to be an approximate linear minimizer.

Comparing to the original Frank-Wolfe method, the idea is that the variant here will hopefully make more progress per iteration, and therefore result in iterates $\boldsymbol{x}$ being combinations of even fewer atoms (i.e. better sparsity). This however comes at a price, namely that the internal problem in each iteration can now become as hard to solve as the original optimization problem, implying that no global run-time guarantees can be given for Algorithm 4 in general.

In computational geometry, the fully corrective method has been used to prove existence results for coresets, e.g. for the smallest enclosing ball problem. Here it is known that compared to the cheaper Algorithm 1, it gives coresets of roughly half the size (Clarkson, 2010). Algorithm 4 is very close to *orthogonal matching pursuit* (Tropp & Gilbert, 2007), which is among the most popular algorithms in signal processing (the difference being that the later applies to an unconstrained domain, more similar to the Frank-Wolfe variant of (Zhang et al., 2012; Harchaoui et al., 2012)). Algorithm 4 for the case of quadratic objectives has also been known as the *minimum-norm-point algorithm* (Bach, 2011). Recently, Yuan & Yan (2012) suggested the use of Newton-type heuristics to solve the subproblems in Algorithm 4.

Away-Steps. Another important variant is the use of *away-steps*, as explained in (GuéLat & Marcotte, 1986), which we can unfortunately not discuss in detail here due to the lack of space. The idea is that in each iteration, we not only add a new atom $\boldsymbol{s}$, but potentially also remove an old atom (provided it is bad with respect to our objective). This requires that the iterate $\boldsymbol{x}$ is represented as a convex combination of the current atoms. Similarily as for the fully corrective Algorithm 4 above, this variant can improve the sparsity of the iterates (Clarkson, 2010). Using away-steps, a faster linear convergence can be obtained for some special problem class (GuéLat & Marcotte, 1986).

The Curvature. The convergence analysis of Frank-Wolfe type algorithms crucially relies on a measure of "non-linearity" of our objective function f over the domain $\mathcal{D}$. The *curvature constant* C_f of a convex and differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, with respect to a compact domain $\mathcal{D}$ is defined as

$$C_f := \sup_{\substack{\boldsymbol{x}, \boldsymbol{s} \in \mathcal{D}, \\ \gamma \in [0,1], \\ \boldsymbol{y} = \boldsymbol{x} + \gamma(\boldsymbol{s} - \boldsymbol{x})}} \frac{2}{\gamma^2}\big(f(\boldsymbol{y}) - f(\boldsymbol{x}) - \langle \boldsymbol{y} - \boldsymbol{x}, \nabla f(\boldsymbol{x}) \rangle\big)\ . \quad (3)$$

For linear functions f for example, it holds that $C_f = 0$. A motivation to consider this quantity follows if we imagine moving from a current point $\boldsymbol{x}$ towards a next "iterate" $\boldsymbol{y} := \boldsymbol{x} + \gamma(\boldsymbol{s} - \boldsymbol{x})$, for any relative "step-size" $\gamma \in [0, 1]$. Bounded C_f then means that the deviation of f at $\boldsymbol{y}$ from the linearization of f given by $\nabla f(\boldsymbol{x})$ at $\boldsymbol{x}$ is bounded, where the acceptable deviation is weighted by the inverse of the squared step-size γ. The defining term $f(\boldsymbol{y}) - f(\boldsymbol{x}) - \langle \boldsymbol{y} - \boldsymbol{x}, \nabla f(\boldsymbol{x})\rangle$ is also widely known as the *Bregman divergence* defined by f. For $f(\boldsymbol{x}) := \frac{1}{2}\|\boldsymbol{x}\|_2^2$ on $\mathbb{R}^n$, the curvature C_f becomes the squared Euclidean diameter of the domain $\mathcal{D}$.

The assumption of bounded curvature C_f closely corresponds to a Lipschitz assumption on the gradient of f (sometimes called C_f*-strong smoothness*). More precisely, if ∇f is L-Lipschitz continuous on $\mathcal{D}$ w.r.t. some arbitrary chosen norm $\|.\|$, then $C_f \leq \operatorname{diam}_{\|.\|}(\mathcal{D})^2 L$, where $\operatorname{diam}_{\|.\|}(.)$ denotes the $\|.\|$-diameter, cf. Appendix D. Note that the curvature constant C_f itself does not depend on the choice of a norm.

Convergence in Primal Error. The following theorem shows that after $O\big(\frac{1}{\varepsilon}\big)$ many iterations, the iterate $\boldsymbol{x}^{(k)}$ of any of the Frank-Wolfe algorithm variants 1, 2, 3, and 4 is an ε-approximate solution to problem (1), i.e. it satisfies $f(\boldsymbol{x}^{(k)}) \leq f(\boldsymbol{x}^*) + \varepsilon$, for $\boldsymbol{x}^*$ being an optimal solution.
Compared to the existing literature (Dunn & Harshbarger, 1978; Jones, 1992; Patriksson, 1993; Zhang, 2003; Clarkson, 2010), our following proof more clearly highlights the dependence on the approximation quality δ of the linear subproblems, holds for all algorithm variants, and will prepare us for the main result of convergence in the duality gap in the next Section. Later we will also show that the resulting convergence rate is indeed best possible for *any* algorithm that adds only one new atom per iteration.

Theorem 1 (Primal Convergence)**.** *For each $k \geq 1$, the iterates $\boldsymbol{x}^{(k)}$ of Algorithms 1, 2, 3, and 4 satisfy*

$$f(\boldsymbol{x}^{(k)}) - f(\boldsymbol{x}^*) \leq \frac{2C_f}{k+2}(1+\delta) \ ,$$

where $\boldsymbol{x}^* \in \mathcal{D}$ is an optimal solution to problem (1), and $\delta \geq 0$ is the accuracy to which the internal linear subproblems are solved (i.e. $\delta = 0$ for Algorithm 1).

The proof of the above convergence theorem relies on expressing the improvement per step in terms of the current duality gap, and then follows along the same idea as in (Clarkson, 2010, Theorem 2.3). A proof is given in Appendix A for completeness.

Inexact Gradient Information. Instead of approximately solving the linear subproblem given by the exact gradient, the same convergence guarantees can be obtained if an inexact gradient is used: For the convergence to hold, we need that the algorithm picks a point $\boldsymbol{s}$ that satisfies $\langle \boldsymbol{s}, d_x\rangle \leq \min_{\boldsymbol{y}\in\mathcal{D}} \langle \boldsymbol{y}, d_x\rangle + \varepsilon'$.

Consider the case when $\mathcal{D}$ is a norm-ball for a norm $\|.\|$, and we only have an estimate $\hat{d}_x$ of the gradient with small $\varepsilon' \geq \left\|\hat{d}_x - d_x\right\|^* \geq \left\|\hat{d}_x\right\|^* - \|d_x\|^* = \max_{\boldsymbol{y}\in\mathcal{D}}\langle \boldsymbol{y}, \hat{d}_x\rangle - \max_{\boldsymbol{y}\in\mathcal{D}}\langle \boldsymbol{y}, d_x\rangle$. This shows that any such minimizer $\boldsymbol{s} := \arg\min_{\boldsymbol{s}\in\mathcal{D}}\langle \boldsymbol{s}, \hat{d}_x\rangle$ is sufficient for the same convergence.

Obtaining a Guaranteed Small Duality Gap. From the above convergence Theorem 1, we have obtained small primal error. However, since the optimum value $f(\boldsymbol{x}^*)$ as well as the curvature constant C_f are often unknown in practical applications, certificates for the current approximation quality are greatly desired. The duality gap $g(\boldsymbol{x})$ that we defined in Section 2 is such an easy computable quality measure, and always upper bounds the primal error $f(\boldsymbol{x}) - f(\boldsymbol{x}^*)$.

Here we state our main result that all variants of the Frank-Wolfe algorithm indeed obtain guaranteed small duality gap $g(\boldsymbol{x}^{(k)}) \leq \varepsilon$ after $O\big(\frac{1}{\varepsilon}\big)$ iterations, over arbitrary bounded domain $\mathcal{D} \subseteq \mathcal{X}$, and even if the linear subproblems are only solved approximately. This generalizes the result of (Clarkson, 2010), which already proved the convergence of Algorithms 1 and 3 on the unit simplex domain (using exact subproblems).

Theorem 2 (Primal-Dual Convergence)**.** *If Algorithm 1, 2, 3 or 4 is run for $K \geq 2$ iterations, then the algorithm has an iterate $\boldsymbol{x}^{(\hat{k})}$, $1 \leq \hat{k} \leq K$, with duality gap bounded by*

$$g(\boldsymbol{x}^{(\hat{k})}) \leq \frac{2\beta C_f}{K+2}(1+\delta) \ ,$$

where $\beta = \frac{27}{8} = 3.375$, and $\delta \geq 0$ is the accuracy to which the linear subproblems are solved.

The proof is provided in Appendix B. The idea is to show that the duality gap cannot stay large over many iterations, since the step improvements would then lead to convergence below the optimal value.

Invariance under Affine Transformations. Interestingly, the Frank-Wolfe algorithm as well as our presented convergence analysis is fully invariant under affine transformations and re-parameterizations of the domain: If we chose any re-parameterization of the domain $\mathcal{D}$, by a *surjective* linear or affine map $M : \hat{\mathcal{D}} \to \mathcal{D}$, then the "old" and "new" optimization problem variants $\min_{\boldsymbol{x}\in\mathcal{D}} f(\boldsymbol{x})$ and $\min_{\hat{\boldsymbol{x}}\in\hat{\mathcal{D}}} \hat{f}(\hat{\boldsymbol{x}})$ for $\hat{f}(\hat{\boldsymbol{x}}) := f(M\hat{\boldsymbol{x}})$ look completely the same to the Frank-Wolfe algorithm: More precisely, every iteration will remain exactly the same, and also the convergence with C_f/k is unchanged, since the curvature constant C_f by its definition (3) is also invariant under such transformations (using that $\nabla\hat{f} = M^T\nabla f$).

A natural variant of such a re-parameterization is the use of *bary-centric coordinates*, if $\mathcal{D}$ is a convex hull of finitely many vectors (then M contains these vectors as columns, and $\hat{\mathcal{D}}$ is the unit simplex). This particularly highlights the importance of the case of simplex

domains, as studied by the seminal paper of (Clarkson, 2010). However, convex hulls of infinitely many vectors can not be represented this way.
The observed invariance under any "distortion" of the domain is surprising in the light of the popularity of pre-conditioners and second-order methods, and the fact that the convergence of the majority of existing convex optimizers crucially depends on the distortion of the domain. Here in contrast, for Frank-Wolfe-type methods, no distortion has any effect.

Optimality in Terms of Sparsity of the Obtained Solutions. We will now show that the number of used atoms (i.e. the sparsity of $\boldsymbol{x}$) of $O\big(\frac{1}{\varepsilon}\big)$ as used by the Frank-Wolfe algorithm is indeed worst-case optimal (for a primal and/or dual approximation error ε), by providing a lower bound of $\Omega\big(\frac{1}{\varepsilon}\big)$. Together with the upper bound, this therefore characterizes the trade-off between *sparsity* and *approximation quality* for the family of optimization problems of the form (1). For the lower bound, the domain is chosen as the unit simplex, $\mathcal{D} := \Delta_n \subseteq \mathbb{R}^n$. The same matching sparsity upper and lower bounds will also hold for optimizing over the ℓ_1-ball instead, and also for the *rank* in trace-norm constrained optimization (Jaggi, 2011).

Consider the function $f(\boldsymbol{x}) := \|\boldsymbol{x}\|_2^2 = \boldsymbol{x}^T\boldsymbol{x}$. Its curvature over the simplex is $C_f = 2\,\mathrm{diam}(\Delta_n)^2 = 4$, which follows directly from the definition (3).

Lemma 3 (see Appendix C)**.** *For* $f(\boldsymbol{x}) := \|\boldsymbol{x}\|_2^2$, *and* $1 \le k \le n$, *it holds that* $\min_{\substack{\boldsymbol{x}\in\Delta_n \\ \mathrm{card}(\boldsymbol{x})\le k}} f(\boldsymbol{x}) = \frac{1}{k}$.

In other words, for any vector $\boldsymbol{x}$ of sparsity $\mathrm{card}(\boldsymbol{x}) = k$, the primal error $f(\boldsymbol{x}) - f(\boldsymbol{x}^*)$ is always lower bounded by $\frac{1}{k} - \frac{1}{n}$. Without considering sparsity, (Canon & Cullum, 1968) have proved a slightly more complicated asymptotic lower bound of $\Omega\big(\frac{1}{k^{1+\mu}}\big)$ on the primal error of the Frank-Wolfe algorithm when run on quadratic objectives, for all $\mu > 0$. Our lower bound here also extends to prove that the obtained duality gap $g(\boldsymbol{x})$ is best possible:

Lemma 4. *For* $f(\boldsymbol{x}) := \|\boldsymbol{x}\|_2^2$, *and any* $k \in \mathbb{N}$, $k < n$, *it holds that* $g(\boldsymbol{x}) \ge \frac{2}{k} \quad \forall \boldsymbol{x} \in \Delta_n$ *s.t.* $\mathrm{card}(\boldsymbol{x}) \le k$.

4. Optimizing over Atomic Sets

For any compact and convex subset $\mathcal{D}$ of a vector space $\mathcal{X}$, the function $\Omega_\mathcal{D} : \mathcal{X} \to \mathbb{R}^+ \cup \{+\infty\}$ defined as

$$\Omega_\mathcal{D}(\boldsymbol{x}) := \inf_{t\ge 0} \{t \mid \boldsymbol{x} \in t\mathcal{D}\}$$

is called the *gauge function* (Rockafellar, 1997) of the convex set $\mathcal{D}$. The *support function* of $\mathcal{D}$ is given by

$$\Omega^*_\mathcal{D}(\boldsymbol{y}) := \sup_{\boldsymbol{s}\in\mathcal{D}} \langle \boldsymbol{s}, \boldsymbol{y}\rangle \ .$$

If the original gauge function $\Omega_\mathcal{D}(.) = \|.\|$ is a norm, then $\Omega^*_\mathcal{D}(.) = \|.\|^*$ is precisely its dual norm.

Atomic Norms. In the special case when the set $\mathcal{D} := \mathrm{conv}(\mathcal{A})$ is a convex hull of another set $\mathcal{A}$, then $\Omega_\mathcal{D}(.)$ becomes the so called *atomic norm* (Chandrasekaran et al., 2012) defined by $\mathcal{A}$. Despite its name, the atomic norm is not always a norm. In general, the function $\Omega_\mathcal{D}(.)$ is known to be a semi-norm if and only if $\mathcal{D}$ is centrally symmetric, and it becomes a norm if $0 \in int(\mathcal{D})$ (Rockafellar, 1997).

The support function of an atomic domain is obtained by taking the largest inner product with an atomic element, $\Omega^*_\mathcal{D}(\boldsymbol{x}) = \sup_{\boldsymbol{s}\in\mathcal{A}}\langle \boldsymbol{s}, \boldsymbol{x}\rangle$, which is often easier to compute than a maximum over the full domain $\mathrm{conv}(\mathcal{A})$. This follows directly from the definition of the convex hull, implying that any linear function attains its maximum over a convex hull at a vertex, or formally $\Omega^*_\mathcal{A}(.) = \Omega^*_{\mathrm{conv}(\mathcal{A})}(.)$. This key property enables the efficient application of the Frank-Wolfe algorithm for atomic domains in the following.

Frank-Wolfe Algorithms for Optimizing over Atomic Domains. In Table 1, we summarize a variety of atomic domains $\mathcal{D}$, over which convex optimization problems of the form (1) can be solved efficiently by the presented Frank-Wolfe methods, using $O\big(\frac{1}{\varepsilon}\big)$ iterations. Depending on the structure of the atoms, this means that the Frank-Wolfe iterates $\boldsymbol{x}$ will often inherit some of this structure, such as sparsity or low rank. In the next subsections we explain these domains more precisely and comment on the computational complexity of the respective linear subproblems. Note that the use of *unit* ball (or gauge) domains comes with no loss of generality, since the argument of f can be re-scaled by an arbitrary constant.

4.1. Optimizing over Vectors

Sparse Vectors / ℓ_1-Ball / Simplex. The convex hull of the signed unit basis vectors $\mathcal{A} = \{\pm\mathbf{e}_i \mid i \in [n]\}$ in $\mathbb{R}^n$ is the unit ball of the ℓ_1-norm. On the other hand, the unit simplex is the convex hull of the unit basis vectors. The use of Frank-Wolfe-type greedy algorithms for finding sparse vectors which optimize a convex function over such domains is well-studied in the literature, see e.g. (Clarkson, 2010) and the references therein. This motivated by the many prominent applications such as for example *Lasso* regression (Tibshirani, 1996), sparse recovery (Mallat & Zhang, 1993), and many learning tasks, where e.g. boosting (Adaboost), support vector machines (Gärtner & Jaggi, 2009; Clarkson, 2010; Ouyang & Gray, 2010), and density estimation (Li & Barron, 2000; Bach et al., 2012) turn out to be such problem instances. Clearly, every

$\mathcal{X}$	**Optimization Domain**		**Complexity of one Frank-Wolfe Iteration**	
	Atoms $\mathcal{A}$	$\mathcal{D} = \operatorname{conv}(\mathcal{A})$	$\Omega^*_{\mathcal{D}}(\boldsymbol{y}) = \sup_{\boldsymbol{s}\in\mathcal{D}}\langle \boldsymbol{s}, \boldsymbol{y}\rangle$	Complexity
$\mathbb{R}^n$	Sparse vectors	$\Vert.\Vert_1$-ball	$\Vert\boldsymbol{y}\Vert_\infty$	$O(n)$
$\mathbb{R}^n$	Sign-vectors	$\Vert.\Vert_\infty$-ball	$\Vert\boldsymbol{y}\Vert_1$	$O(n)$
$\mathbb{R}^n$	ℓ_p-Sphere	$\Vert.\Vert_p$-ball	$\Vert\boldsymbol{y}\Vert_q$	$O(n)$
$\mathbb{R}^n$	Sparse non-neg. vectors	Simplex Δ_n	$\max_i\{\boldsymbol{y}_i\}$	$O(n)$
$\mathbb{R}^n$	Latent group sparse vectors	$\Vert.\Vert_{\mathcal{G}}$-ball	$\max_{g\in\mathcal{G}} \left\Vert\boldsymbol{y}_{(g)}\right\Vert_g^*$	$\sum_{g\in\mathcal{G}} \vert g\vert$
$\mathbb{R}^{m\times n}$	Matrix trace norm	$\Vert.\Vert_{tr}$-ball	$\Vert\boldsymbol{y}\Vert_{op} = \sigma_1(\boldsymbol{y})$	$\tilde{O}\big(N_f/\sqrt{\varepsilon'}\big)$ (Lanczos)
$\mathbb{R}^{m\times n}$	Matrix operator norm	$\Vert.\Vert_{op}$-ball	$\Vert\boldsymbol{y}\Vert_{tr} = \Vert(\sigma_i(\boldsymbol{y}))\Vert_1$	SVD
$\mathbb{R}^{m\times n}$	Schatten matrix norms	$\Vert(\sigma_i(.))\Vert_p$-ball	$\Vert(\sigma_i(\boldsymbol{y}))\Vert_q$	SVD
$\mathbb{R}^{m\times n}$	Matrix max-norm	$\Vert.\Vert_{\max}$-ball		$\tilde{O}\big(N_f(n+m)^{1.5}/\varepsilon'^{2.5}\big)$
$\mathbb{R}^{n\times n}$	Permutation matrices	Birkhoff polytope		$O(n^3)$
$\mathbb{R}^{n\times n}$	Rotation matrices			SVD (Procrustes prob.)
$\mathbb{S}^{n\times n}$	Rank-1 PSD matrices of unit trace	$\{\boldsymbol{x}\succeq 0,\ \operatorname{Tr}(\boldsymbol{x}) = 1\}$	$\lambda_{\max}(\boldsymbol{y})$	$\tilde{O}\big(N_f/\sqrt{\varepsilon'}\big)$ (Lanczos)
$\mathbb{S}^{n\times n}$	PSD matrices of bounded diagonal	$\{\boldsymbol{x}\succeq 0,\ \boldsymbol{x}_{ii}\le 1\}$		$\tilde{O}\big(N_f\, n^{1.5}/\varepsilon'^{2.5}\big)$

Table 1. Some examples of atomic domains suitable for optimization using the Frank-Wolfe algorithm. *Here SVD refers to the complexity of computing a singular value decomposition, which is $O(\min\{mn^2, m^2n\})$. N_f is the number of non-zero entries in the gradient of the objective function f, and $\varepsilon' = \frac{\delta C_f}{k+2}$ is the required accuracy for the linear subproblems. For any $p \in [1,\infty]$, the conjugate value q is meant to satisfy $\frac{1}{p} + \frac{1}{q} = 1$, allowing $q = \infty$ for $p = 1$ and vice versa.*

iteration will add at most one new non-zero coordinate to $\boldsymbol{x}$, and the linear subproblems consist of finding the largest entry of the gradient.

The resulting trade-off between the *sparsity* and the *approximation quality* is interesting. Our above sparsity lower bounds from Lemmata 3 and 4 together with the upper bounds of $O\big(\frac{1}{\varepsilon}\big)$ from the convergence analysis show that the sparsity of the Frank-Wolfe iterates is indeed best possible in terms of both primal and dual approximation quality. For optimizing over the simplex, this trade-off was also described by (Gärtner & Jaggi, 2009; Clarkson, 2010), and by (Shalev-Shwartz et al., 2010) for the ℓ_1-ball (considering primal error).

The ℓ_p-Ball. An exact Frank-Wolfe iteration only costs linear time when optimizing over any ℓ_p-ball domain $\mathcal{D}$, for $p \in [1,\infty]$. This follows by the duality of the ℓ_p and ℓ_q-norms, as in Hölder's inequality $\langle \boldsymbol{s}, \boldsymbol{y}\rangle \le \Vert\boldsymbol{s}\Vert_p \cdot \Vert\boldsymbol{y}\Vert_q$ (for $p, q \in [1,\infty]$, $\frac{1}{p} + \frac{1}{q} = 1$, allowing $q = \infty$ for $p = 1$ and vice versa). An optimal solution $\boldsymbol{s}$ to the linear problem $\max_{\hat{\boldsymbol{s}},\, \Vert\hat{\boldsymbol{s}}\Vert_p \le 1} \hat{\boldsymbol{s}}^T\boldsymbol{y}$ can simply be obtained from $\boldsymbol{y}$ by choosing $|\boldsymbol{s}_i| \propto |\boldsymbol{y}_i|^{q-1}$, keeping the same signs. This also holds for the case $p = \infty$, $q = 1$, where the domain $\mathcal{D}$ becomes the cube.

Structured Atomic Norms. In recent years, structured norms have gained strong interest in several areas of machine learning, computer vision, and signal processing, due to their ability to induce more general and structured notions of sparsity, see e.g. (Jenatton et al., 2011) for an overview.

Here we will focus on one large class of structured norms, proposed by (Obozinski et al., 2011), which due to the atomic structure is particularly suitable to be used with the Frank-Wolfe algorithm. Let $\mathcal{G}$ be a finite collection of groups of indices $g \subseteq [n]$ (which are allowed to overlap), and $\bigcup_{g\in\mathcal{G}} g = [n]$. For each group g, we choose an arbitrary norm $\Vert.\Vert_g$, which acts only on the coordinates belonging to g, i.e. on $\mathbb{R}^{|g|}$. For any $\boldsymbol{v} \in \mathbb{R}^n$ and $g \subseteq [n]$, we write $\boldsymbol{v}_{[g]} \in \mathbb{R}^n$ for the vector coinciding with $\boldsymbol{v}$ in the coordinates in g, and being zero elsewhere, i.e. $\operatorname{supp}(\boldsymbol{v}_{[g]}) \subseteq g$. The same vector when restricted to these coordinates is written as $\boldsymbol{v}_{(g)} \in \mathbb{R}^{|g|}$. In this setting, a slight generalization of the *latent group norm* (Obozinski et al., 2011) is given by

$$\begin{aligned}\Vert\boldsymbol{x}\Vert_{\mathcal{G}} := \min_{\boldsymbol{v}_{(g)}\in\mathbb{R}^{|g|}} \quad & \textstyle\sum_{g\in\mathcal{G}} \left\Vert\boldsymbol{v}_{(g)}\right\Vert_g \\ s.t. \quad & \boldsymbol{x} = \textstyle\sum_{g\in\mathcal{G}} \boldsymbol{v}_{[g]}\ .\end{aligned}$$

It is known (Obozinski et al., 2011) that this norm is an atomic norm (and a norm), with the atoms $\mathcal{A} = \{\mathbb{D}_g \,|\, g \in \mathcal{G}\}$ being the unit disks defined by the norms on the groups, $\mathbb{D}_g := \left\{\boldsymbol{v} \in \mathbb{R}^n \,\middle|\, \substack{\operatorname{supp}(\boldsymbol{v})\subseteq g,\\ \Vert\boldsymbol{v}_{(g)}\Vert_g \le 1}\right\}$. As we discussed above when introducing atomic norms, this implies that the dual norm is now given by $\Vert\boldsymbol{y}\Vert^*_{\mathcal{G}} = \max_{g\in\mathcal{G}} \left\Vert\boldsymbol{y}_{(g)}\right\Vert^*_g$. Furthermore, we can intersect any such atomic set of disks with the non-negative cone, and therefore obtain a corresponding "non-negative" atomic norm. In the special case that $\mathcal{G}$ forms a partition of $[n]$, and all group norms $\Vert.\Vert_g$ are chosen as the Euclidian norm, then $\Vert.\Vert_{\mathcal{G}}$ becomes the standard group-lasso penalty (Yuan & Lin, 2006).

4.2. Optimizing over Matrices

Schatten Matrix Norms. If $\Vert.\Vert$ is a vector norm on $\mathbb{R}^r$, $r := \min\{m, n\}$, then the corresponding *Schatten matrix norm* of a matrix $M \in \mathbb{R}^{m\times n}$ is defined as $\Vert(\sigma_1(M), \dots, \sigma_r(M))\Vert$, where $\sigma_1(M), \dots, \sigma_r(M)$ are the singular values of M. The dual of the Schatten ℓ_p-norm is the Schatten ℓ_q-norm. The two most prominent examples are the trace norm $\|.\|_{tr}$ (also called the nuclear- or Schatten ℓ_1-norm, being the sum of the singular values), and the operator norm $\|.\|_{op}$ (Schatten ℓ_∞-norm, being the largest singular value).

To apply the Frank-Wolfe algorithm to minimize a convex function over a norm ball of a Schatten-ℓ_p-norm, we need to be able to solve the linear subproblems of the form $\sup_{S \in \mathcal{D}} \langle S, M \rangle$. Here, the following fact comes to help: Since Schatten norms are invariant under orthogonal transformations (by invariance of the spectrum of the matrix), we can find such minimizers by employing the singular value decomposition (SVD): If the SVD of the given matrix $M \in \mathbb{R}^{m \times n}$ is $U \operatorname{diag}(\boldsymbol{\sigma}) V^T = M$ (where $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_r) \in \mathbb{R}^r$ and $U \in \mathbb{R}^{m \times r}$, $V \in \mathbb{R}^{n \times r}$ are orthonormal), then $S := U \operatorname{diag}(\boldsymbol{s}) V^T$ is an optimizer of $\sup_{S \in \mathcal{D}} \langle S, M \rangle$, if $\boldsymbol{s}$ is any vector attaining $\boldsymbol{s}^T \boldsymbol{\sigma} = \|\boldsymbol{\sigma}\|_q$ with $\|\boldsymbol{s}\|_p \leq 1$. While finding such a conjugate vector $\boldsymbol{s}$ only costs linear time $O(r)$, the main computational cost of a Frank-Wolfe step on a Schatten norm domain remains the computation of the SVD (of the current gradient matrix M), which is in $O(\min\{mn^2, m^2 n\})$.
In the important case of optimizing over bounded *trace-norm* (Schatten ℓ_1-norm), the subproblems can be solved much more efficiently, by a single approximate eigenvector computation instead of a complete SVD. We discuss this case in more detail in Section 4.3.

Orthonormal Matrices, and the Operator Norm Ball. The convex hull of all orthonormal matrices $U \in \mathbb{R}^{m \times n}$, $U^T U = \mathbf{I}$, is the norm ball of the standard matrix operator norm $\|.\|_{op}$ on $\mathbb{R}^{m \times n}$, which is the Schatten-ℓ_∞-norm. Here it becomes particularly easy to obtain a linear optimizer over $\mathcal{D}$ (the operator norm ball) using the SVD approach we have explained above for general Schatten norms. If $U \Sigma V^T = M$ is the SVD of M, then $S := UV^T$ is a solution to the linear problem $\sup_{\|S\|_{op} \leq 1} \langle S, M \rangle$. (So for $p = \infty$, $\operatorname{diag}(\boldsymbol{s})$ is always the identity matrix).

Permutation Matrices. The convex hull of all $n \times n$ permutation matrices is known as the Birkhoff polytope, and coincides with the set of all doubly stochastic matrices (Lovász & Plummer, 2009). Despite the number of atoms being exponential ($n!$), a linear function can be optimized efficiently over this polytope, by using the primal-dual Hungarian algorithm in time $O(n^3)$ (Lovász & Plummer, 2009). Therefore, the exact Frank-Wolfe algorithm can be applied efficiently for such domains, see also (Tewari et al., 2011).

Rotation Matrices. We consider optimizing over the convex hull of all rotation matrices, i.e. the orthogonal $n \times n$ matrices of determinant one. Linear optimization over this set $\mathcal{D}$ is known as the *orthogonal Procrustes problem*, and can be solved by one SVD. We can therefore optimize arbitrary convex functions f by the Frank-Wolfe algorithm, using combinations of only few rotations matrices. An online-version of such optimization tasks was studied in (Hazan et al., 2010).

4.3. Factorized Matrix Norms

In this section, we propose a new general framework for optimization over factorizations of a matrix $M \in \mathbb{R}^{m \times n}$ into two factors $M = LR^T$, where $L \in \mathbb{R}^{m \times r}$, $R \in \mathbb{R}^{n \times r}$ for some $r \in \mathbb{N}$. To do so, we consider atomic domains which consist of the (matrix) *outer products of two atomic sets*, i.e.

$$\mathcal{A} := \left\{ LR^T \;\middle|\; \substack{L \in \mathcal{A}_{\text{left}} \,, \\ R \in \mathcal{A}_{\text{right}}} \right\} ,$$

where $\mathcal{A}_{\text{left}} \subseteq \mathbb{R}^{m \times r}$ and $\mathcal{A}_{\text{right}} \subseteq \mathbb{R}^{n \times r}$ are arbitrary compact subsets (not necessarily finite) of $\mathbb{R}^{m \times r}$ and $\mathbb{R}^{n \times r}$ respectively, and r is fixed.

By definition of this atomic set, any iteration of the Frank-Wolfe algorithm when optimizing over $\mathcal{D} = \operatorname{conv}(\mathcal{A})$ will result in an update of the form $\boldsymbol{s} = LR^T$, that is a *low-rank* update (of rank $\leq r$). In other words, such domains allow us to maintain all Frank-Wolfe iterates $\boldsymbol{x}$ as a low-rank matrix factorization (of rank at most $\leq rk$ in step k).

Our definition can also be seen as a generalization of the fact that any pair of norms on vectors $\boldsymbol{u} \in \mathbb{R}^m$ and $\boldsymbol{v} \in \mathbb{R}^n$ does induce a matrix norm on $\mathbb{R}^{m \times n}$, by means of the quadratic form $\boldsymbol{u}^T M \boldsymbol{v}$, see e.g. (Bach et al., 2008; Zhang et al., 2012) and (Boyd & Vandenberghe, 2004, Example 3.11). We recover this case when $r = 1$. (The work of Zhang et al. (2012) appeared after our paper was put online).

Trace Norm. The trace norm (Schatten ℓ_1-norm) gives the most natural example of such a factorized matrix norm. The unit ball of the trace norm is known to be the convex hull of the rank-1 matrices $\mathcal{A} := \left\{ \boldsymbol{u}\boldsymbol{v}^T \;\middle|\; \substack{\boldsymbol{u} \in \mathbb{R}^n,\, \|\boldsymbol{u}\|_2 = 1 \\ \boldsymbol{v} \in \mathbb{R}^m,\, \|\boldsymbol{v}\|_2 = 1} \right\}$. Here, compared to the cubic complexity of solving the linear subproblem for general Schatten norms (using SVD, as explained in Section 4.2), the Frank-Wolfe steps become much more efficient. This is because the subproblem amounts to approximating the top eigenvalue (or singular value), which when using the standard Lanczos' algorithm takes $\tilde{O}(N_f / \sqrt{\varepsilon})$ arithmetic operations (suppressing constants and logarithmic factors), see e.g. Appendix E, when N_f is the number of non-zeros of ∇f. Altogether, the Frank-Wolfe algorithm therefore provides ε-accurate low-rank solutions (rank $O\left(\frac{1}{\varepsilon}\right)$) in a total running time of $\tilde{O}\big(N_f / \varepsilon^{1.5}\big)$, which is near-linear in the number of non-zeros N_f, see (Jaggi & Sulovský, 2010). This contrasts the accelerated versions of the "singular value thresholding" algorithm

r	$\mathcal{A}_{\text{left}} \subseteq \mathbb{R}^{m\times r}$	$\mathcal{A}_{\text{right}} \subseteq \mathbb{R}^{n\times r}$	$\Omega_{\text{conv}(\mathcal{A})}(M)$	$\Omega^*_{\mathcal{A}}(M)$	**FW step**
1	$\|.\|_2$-sphere	$\|.\|_2$-sphere	Trace norm $\|M\|_{tr}$	$\|M\|_{op}$	Lanczos, see Table 1
1	$\|.\|_1$-sphere	$\|.\|_1$-sphere	Vector ℓ_1-norm $\|\vec{M}\|_1$	$\|\vec{M}\|_\infty$	$O(nm)$
1	$\|.\|_\infty$-sphere	$\|.\|_\infty$-sphere		Cut-norm $\|.\|_{\infty\to 1}$	NP-hard (Alon & Naor, 2006)
n+m	$\|.\|_{2,\infty}$	$\|.\|_{2,\infty}$	Max-norm $\|M\|_{\max}$		SDP, see Table 1
1	$\|.\|_2 \cap \mathbb{R}^m_{\geq 0}$	$\|.\|_2 \cap \mathbb{R}^n_{\geq 0}$	"non-neg. trace norm"		NP-hard (Murty & Kabadi, 1987)
1	Simplex Δ_m	Simplex Δ_n	"non-neg. matrix ℓ_1-norm"		$O(nm)$

Table 2. Examples of some factorized matrix norms on $\mathbb{R}^{m\times n}$, each induced by two atomic norms (last two rows giving non-negative factorizations). Here $\|M\|_{2,\infty}$ is the length of the ℓ_2-longest row of the matrix M, and $\vec{M}$ denotes the entries of M written in a single large vector.

of (Cai et al., 2010), which perform $O(1/\sqrt{\varepsilon})$ complete SVD computations, in each iteration taking time cubic in the matrix dimension.

For trace-norm optimization, the presentation here avoids the detour over a semidefinite programming formulation present in (Jaggi & Sulovský, 2010) when applying the method of (Hazan, 2008). The same algorithm applies to optimizing under constrained *weighted* trace norm, by reduction to the trace-norm as e.g. described in (Giesen et al., 2012). For optimizing over semidefinite matrices $\mathbb{S}^{n\times n}$ of bounded trace, the above discussion is analogous, with $\mathcal{A} := \big\{uu^T \;\big|\; \boldsymbol{u} \in \mathbb{R}^n,\ \|\boldsymbol{u}\|_2 = 1\big\}$.

General Factorized Matrix Norm Domains. Even in the case when optimizing over the individual atomic domains (given by $\mathcal{A}_{\text{left}}$ and $\mathcal{A}_{\text{right}}$) is easy, optimizing a linear function over such a product domain $\mathcal{A}$ can rapidly turn into an intractable combinatorial problem. For example, maximizing $\langle \boldsymbol{u}\boldsymbol{v}^T, M\rangle$ over vectors $\|\boldsymbol{u}\|_\infty \leq 1$ and $\|\boldsymbol{v}\|_\infty \leq 1$ for a given matrix M amounts to computing the *cut-norm* $\|M\|_{\infty\to 1}$, which is NP-hard (Alon & Naor, 2006). Maximizing the same quadratic form over non-negative vectors $\|\boldsymbol{u}\|_2 \leq 1$, $\boldsymbol{u} \geq 0$ and $\|\boldsymbol{v}\|_2 \leq 1$, $\boldsymbol{v} \geq 0$ was also shown to be NP-hard by (Murty & Kabadi, 1987).

Matrix Max-Norm, and Semidefinite Optimization with Bounded Diagonal. Another efficiently tractable case of a factorized matrix domain is given by the matrix max-norm, which is known to be an approximation of the cut-norm (Srebro & Shraibman, 2005). Optimizing a linear function over the PSD matrices with all diagonal elements upper bounded by one is a well-studied problem, e.g. appearing as the standard SDP relaxation of the Max-Cut problem (Goemans & Williamson, 1995). The algorithm of (Arora et al., 2005) delivers an additive ε'-approximation to the linearized problem over such matrices in time $\tilde{O}\left(\frac{n^{1.5}L^{2.5}}{\varepsilon'^{2.5}}N_M\right)$ where $L > 0$ is an upper bound on the value of the linear problem, and N_M is the number of non-zeros in M (Jaggi, 2011, Section 3.5).

Using the alternative characterization of the *max-norm* of a rectangular matrix $M \in \mathbb{R}^{m\times n}$ in terms of a semidefinite program of the above form (Srebro & Shraibman, 2005; Jaggi, 2011), we can directly plug in the algorithm of (Arora et al., 2005) into the Frank-Wolfe method, in order to optimize any convex function over a max-norm constrained domain. This, to our knowledge, gives the first algorithm with a convergence guarantee for such problems. (Lee et al., 2010) have studied a proximal optimizer on a non-convex formulation of the max-norm, and very recently, (Orabona et al., 2012) have introduced a first-order smoothing technique for max-norm problems.

4.4. Optimizing over Submodular Polyhedra

For a finite ground set S, a real valued function defined on all subsets of S, is called *submodular*, if $g(A \cap B) + g(A \cup B) \;\leq\; g(A) + g(B)$ holds $\forall A, B \subseteq S$. For any given submodular function g with $g(\emptyset) = 0$, the corresponding *submodular polyhedron* (or polymatroid) is defined as the convex set $\mathcal{P}_g := \big\{x \in \mathbb{R}^n \;\big|\; \sum_{i\in A} x_i \leq g(A) \;\; \forall A \subseteq S\big\}$, where $n = |S|$.

Our presented Frank-Wolfe algorithm variants directly apply to minimization of a convex function f over such a domain. This follows since linear optimization over such a submodular polyhedron domain is efficient, by an $O(n\log n)$ time greedy algorithm (Edmonds, 1970; Lovász, 1983; Bach, 2011). (Note that for compactness, the domain is usually restricted to the non-negative orthant $\mathcal{D} := \mathcal{P}_g \cap \mathbb{R}^n_{\geq 0}$). Submodular optimization is currently gaining increased interest as a more general way to relate combinatorial problems to convexity, such as for example for structured sparsity, see e.g. (Bach, 2011).

Acknowledgements. The author would like to thank Alexandre d'Aspremont, Francis Bach, Guillaume Bouchard, Bernd Gärtner, Zaid Harchaoui, Stefanie Jegelka, Rodolphe Jenatton, Andreas Krause, Simon Lacoste-Julien and Guillaume Obozinski for helpful discussions and remarks, and Robert Carnecky for the 3d-visualization. MJ acknowledges support by the ERC Project SIPA, by a Google research award, and by the Swiss National Science Foundation (SNSF). Most of this work was done while at ETH Zurich.

References

Alon, N and Naor, A. Approximating the Cut-Norm via Grothendieck's Inequality. *SIAM J. Computing*, 2006.

Arora, S, Hazan, E, and Kale, S. Fast algorithms for approximate semidefinite programming using the multiplicative weights update method. *FOCS*, 2005.

Bach, F. Learning with Submodular Functions: A Convex Optimization Perspective. 2011.

Bach, F, Mairal, J, and Ponce, J. Convex Sparse Matrix Factorizations. Technical report, 2008.

Bach, F, Lacoste-Julien, S, and Obozinski, G. On the Equivalence between Herding and Conditional Gradient Algorithms. In *ICML*, 2012.

Boyd, S and Vandenberghe, L. *Convex optimization*. 2004.

Cai, J-F, Candes, E J, and Shen, Z. A Singular Value Thresholding Algorithm for Matrix Completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

Canon, M D and Cullum, C D. A Tight Upper Bound on the Rate of Convergence of Frank-Wolfe Algorithm. *SIAM Journal on Control*, 6(4):509–516, 1968.

Chandrasekaran, V, Recht, B, Parrilo, P A, and Willsky, A S. The Convex Geometry of Linear Inverse Problems. *Found. Comp. Math.*, 12(6):805–849, 2012.

Clarkson, K L. Coresets, Sparse Greedy Approximation, and the Frank-Wolfe Algorithm. *ACM Transactions on Algorithms*, 6(4), 2010.

Demyanov, V F and Rubinov, A M. *Approximate methods in optimization problems*. Elsevier, 1970.

Dudik, M, Harchaoui, Z, and Malick, J. Lifted coordinate descent for learning with trace-norm regularization. In *AISTATS*, 2012.

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

Edmonds, J. Submodular Functions, Matroids, and Certain Polyhedra. In *Comb. Struct. and Appl.*, 69–87, 1970.

Frank, M and Wolfe, P. An algorithm for quadratic programming. *Naval Res. Logis. Quart.*, 3:95–110, 1956.

Gärtner, B and Jaggi, M. Coresets for polytope distance. *ACM SCG*, 2009.

Giesen, J, Jaggi, M, and Laue, S. Regularization Paths with Guarantees for Convex Semidefinite Optimization. *AISTATS*, 2012.

Goemans, M and Williamson, D. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6), 1995.

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

Harchaoui, Z, Juditsky, A, and Nemirovski, A. Conditional gradient algorithms for machine learning. In *NIPS Workshop on Optimization for ML*, December 2012.

Hazan, E. Sparse Approximate Solutions to Semidefinite Programs. In *LATIN*, pp. 306–316, 2008.

Hazan, E and Kale, S. Projection-free Online Learning. In *ICML*, 2012.

Hazan, E, Kale, S, and Warmuth, M.K. Learning rotations with little regret. *In COLT*, pp. 144–154, 2010.

Jaggi, M. *Sparse Convex Optimization Methods for Machine Learning*. PhD thesis, ETH Zürich, 2011.

Jaggi, M and Sulovský, M. A Simple Algorithm for Nuclear Norm Regularized Problems. *ICML*, 2010.

Jenatton, R, Audibert, J-Y, and Bach, F. Structured Variable Selection with Sparsity-Inducing Norms. *JMLR*, 12: 2777–2824, 2011.

Jones, L K. A Simple Lemma on Greedy Approximation in Hilbert Space and Convergence Rates for Projection Pursuit Regression and Neural Network Training. *The Annals of Statistics*, 20(1):608–613, 1992.

Kuczyński, J and Woźniakowski, H. Estimating the Largest Eigenvalue by the Power and Lanczos Algorithms with a Random Start. *SIAM Journal on Matrix Analysis and Applications*, 13(4):1094–1122, 1992.

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

Lee, J, Recht, B, Salakhutdinov, R, Srebro, N, and Tropp, J A. Practical Large-Scale Optimization for Max-Norm Regularization. *NIPS*, 2010.

Levitin, E S and Polyak, B T. Constrained minimization methods. *USSR Comp. Math. & M. Phys.*, 6(5), 1966.

Li, J and Barron, A. Mixture density estimat.. *NIPS*, 2000.

Lovász, L. Submodular functions and convexity. *Mathematical programming: the state of the art*, 1983.

Lovász, L and Plummer, M D. *Matching Theory*. American Mathematical Society, 2009.

Mallat, S G and Zhang, Z. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

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

Nesterov, Y. *Introductory Lectures on Convex Optimization*. A Basic Course. Kluwer, 2004.

Obozinski, G, Jacob, L, and Vert, JP. Group Lasso with Overlaps: the Latent Group Lasso approach. *arXiv*, 2011.

Orabona, F, Argyriou, A, and Srebro, N. PRISMA: PRoximal Iterative SMoothing Algorithm. *arXiv.org*, 2012.

Ouyang, H. and Gray, A. Fast Stochastic Frank-Wolfe Algorithms for Nonlinear SVMs. *SDM*, 2010.

Patriksson, M. Partial linearization methods in nonlinear programming. *Journal of Optimization Theory and Applications*, 78(2):227–246, 1993.

Rockafellar, R T. *Convex analysis*. 1997.

Shalev-Shwartz, S, Srebro, N, and Zhang, T. Trading Accuracy for Sparsity in Optimization Problems with Sparsity Constraints. *SIAM J. on Optimization*, 20, 2010.

Srebro, N and Shraibman, A. Rank, Trace-Norm and Max-Norm. In *COLT*, 545–560, 2005.

Temlyakov, V N. Greedy approximation in convex optimization. *arXiv.org*, stat.ML, 2012.

Tewari, A, Ravikumar, P, and Dhillon, I S. Greedy Algorithms for Structurally Constrained High Dimensional Problems. In *NIPS*, 2011.

Tibshirani, R. Regression Shrinkage and Selection via the Lasso. *J. Royal Statistical Society. Series B*, 1996.

Tropp, J A and Gilbert, A. Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit. *IEEE Trans. on Information Theory*, 53(12):4655–4666, 2007.

Yuan, M and Lin, Y. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, 68(1):49–67, 2006.

Yuan, X-T and Yan, S. Forward Basis Selection for Sparse Approximation over Dictionary. In *AISTATS*, 2012.

Zhang, T. Sequential greedy approximation for certain convex optimization problems. *IEEE Transactions on Information Theory*, 49(3):682–691, 2003.

Zhang, X, Yu, Y, and Schuurmans, D. Accelerated Training for Matrix-norm Regularization: A Boosting Approach. In *NIPS*, 2012.

A. Primal Convergence

The proof of the convergence rate of the primal error crucially depends on the following Lemma 5 on the improvement in each iteration, expressing this improvement in terms of the current duality gap. Using the lemma, the convergence proof then follows along the same idea as in (Clarkson, 2010, Theorem 2.3). Note that a weaker variant of Lemma 5 for the exact case $\delta = 0$ was already proven by (Frank & Wolfe, 1956) (without allowing for approximate linear minimizers).

Lemma 5. *For a step $\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)} + \gamma(\boldsymbol{s} - \boldsymbol{x}^{(k)})$ with arbitrary step-size $\gamma \in [0, 1]$, it holds that*

$$f(\boldsymbol{x}^{(k+1)}) \le f(\boldsymbol{x}^{(k)}) - \gamma g(\boldsymbol{x}^{(k)}) + \tfrac{\gamma^2}{2} C_f (1 + \delta) \ ,$$

if $\boldsymbol{s}$ is an approximate linear minimizer, i.e. $\langle \boldsymbol{s}, \nabla f(\boldsymbol{x}^{(k)}) \rangle \le \min_{\hat{\boldsymbol{s}} \in \mathcal{D}} \langle \hat{\boldsymbol{s}}, \nabla f(\boldsymbol{x}^{(k)}) \rangle + \frac{1}{2}\delta\gamma C_f$.

Proof. We write $\boldsymbol{x} := \boldsymbol{x}^{(k)}$, $\boldsymbol{y} := \boldsymbol{x}^{(k+1)} = \boldsymbol{x} + \gamma(\boldsymbol{s} - \boldsymbol{x})$, and $d_x := \nabla f(\boldsymbol{x})$ to simplify the notation. From the definition of the curvature constant C_f of our convex function f, we have

$$\begin{aligned} f(\boldsymbol{y}) = &\ f(\boldsymbol{x} + \gamma(\boldsymbol{s} - \boldsymbol{x})) \\ \le &\ f(\boldsymbol{x}) + \gamma \langle \boldsymbol{s} - \boldsymbol{x}, d_x \rangle + \tfrac{\gamma^2}{2} C_f \ . \end{aligned}$$

Now we use that the choice of $\boldsymbol{s}$ is a good "descent direction" on the linear approximation to f at $\boldsymbol{x}$. Formally, we are given a point $\boldsymbol{s}$ that satisfies $\langle \boldsymbol{s}, d_x \rangle \le \min_{\boldsymbol{y} \in \mathcal{D}} \langle \boldsymbol{y}, d_x \rangle + \frac{1}{2}\delta\gamma C_f$, or in other words

$$\begin{aligned} \langle \boldsymbol{s} - \boldsymbol{x}, d_x \rangle \le &\ \min_{\boldsymbol{y} \in \mathcal{D}} \langle \boldsymbol{y}, d_x \rangle - \langle \boldsymbol{x}, d_x \rangle + \tfrac{1}{2}\delta\gamma C_f \\ = &\ -g(\boldsymbol{x}) + \tfrac{1}{2}\delta\gamma C_f \ . \end{aligned}$$

Here we have plugged in the definition (2) of the duality gap $g(\boldsymbol{x})$. Altogether, we therefore obtain $f(\boldsymbol{y}) \le f(\boldsymbol{x}) - \gamma g(\boldsymbol{x}) + \frac{\gamma^2}{2} C_f (1 + \delta)$, which proves the lemma. □

Theorem' 1 (Primal Convergence). *For each $k \ge 1$, the iterates $\boldsymbol{x}^{(k)}$ of Algorithms 1, 2, 3, and 4 satisfy*

$$f(\boldsymbol{x}^{(k)}) - f(\boldsymbol{x}^*) \le \frac{2C_f}{k+2}(1 + \delta) \ ,$$

where $\boldsymbol{x}^* \in \mathcal{D}$ is an optimal solution to problem (1), and $\delta \ge 0$ is the accuracy to which the internal linear subproblems are solved (in the exact Algorithm 1, we have $\delta = 0$).

Proof. From Lemma 5 we know that for every step of Algorithm 2, it holds that $f(\boldsymbol{x}^{(k+1)}) \le f(\boldsymbol{x}^{(k)}) - \gamma g(\boldsymbol{x}^{(k)}) + \gamma^2 C$, if we define $C := \frac{C_f}{2}(1 + \delta)$. For the line-search variant as in Algorithm 3 and for the "fully corrective" Algorithm 4, the same bound (using the same fixed $\gamma := \frac{2}{k+2}$ on the right-hand side) also holds, simply by inclusion of the fixed step-size case in the respective minimum, i.e. $f(\boldsymbol{x}^{(k+1)}_{\text{Re-Opt}}) \le f(\boldsymbol{x}^{(k+1)}_{\text{Line-Search}}) \le f(\boldsymbol{x}^{(k+1)}_{\gamma})$. For the exact variant (i.e. Algorithm 1), the bound holds for $\delta = 0$.

Writing $h(\boldsymbol{x}) := f(\boldsymbol{x}) - f(\boldsymbol{x}^*)$ for the (unknown) primal error at any point $\boldsymbol{x}$, this implies that

$$\begin{aligned} h(\boldsymbol{x}^{(k+1)}) \le &\ h(\boldsymbol{x}^{(k)}) - \gamma g(\boldsymbol{x}^{(k)}) + \gamma^2 C \\ \le &\ h(\boldsymbol{x}^{(k)}) - \gamma h(\boldsymbol{x}^{(k)}) + \gamma^2 C \\ = &\ (1 - \gamma) h(\boldsymbol{x}^{(k)}) + \gamma^2 C \ , \end{aligned} \tag{4}$$

where we have used weak duality $h(\boldsymbol{x}) \le g(\boldsymbol{x})$ as discussed after the definition of the duality gap (2). We will now use induction over k to prove our claimed bound, i.e.

$$h(\boldsymbol{x}^{(k+1)}) \le \tfrac{4C}{k+1+2} \qquad k = 0, 1, \ldots$$

The base-case $k = 0$ follows from (4) applied for the first step of the algorithm, using $\gamma = \gamma^{(0)} = \frac{2}{0+2} = 1$.

Now considering $k \ge 1$, the bound (4) reads as

$$\begin{aligned} h(\boldsymbol{x}^{(k+1)}) \le &\ (1 - \gamma^{(k)}) h(\boldsymbol{x}^{(k)}) + {\gamma^{(k)}}^2 C \\ = &\ (1 - \tfrac{2}{k+2}) h(\boldsymbol{x}^{(k)}) + \left(\tfrac{2}{k+2}\right)^2 C \\ \le &\ (1 - \tfrac{2}{k+2}) \tfrac{4C}{k+2} + \left(\tfrac{2}{k+2}\right)^2 C \ , \end{aligned}$$

where in the last inequality we have used the induction hypothesis for $\boldsymbol{x}^{(k)}$. Simply rearranging the terms gives

$$\begin{aligned} h(\boldsymbol{x}^{(k+1)}) \le &\ \tfrac{4C}{k+2}\left(1 - \tfrac{1}{k+2}\right) = \tfrac{4C}{k+2} \tfrac{k+2-1}{k+2} \\ \le &\ \tfrac{4C}{k+2} \tfrac{k+2}{k+3} = \tfrac{4C}{k+3} \ , \end{aligned}$$

which is our claimed bound for $k \ge 1$. □

B. Primal-Dual Convergence

Theorem' 2 (Primal-Dual Convergence). *If Algorithm 1, 2, 3 or 4 is run for $K \ge 2$ iterations, then the algorithm has an iterate $\boldsymbol{x}^{(\hat{k})}$, $1 \le \hat{k} \le K$, with duality gap bounded by*

$$g(\boldsymbol{x}^{(\hat{k})}) \le \frac{2\beta C_f}{K+2}(1 + \delta) \ ,$$

where $\beta = \frac{27}{8} = 3.375$, and $\delta \ge 0$ is the accuracy to which the linear subproblems are solved.

Proof. We will actually prove that the iterate of small duality gap will appear in the last third of the K iterations. To simplify notation, we will denote the primal and dual errors for any iteration $k \ge 0$ in the algorithm by $h^{(k)} := h(\boldsymbol{x}^{(k)})$ and $g^{(k)} := g(\boldsymbol{x}^{(k)})$.

By our previous primal convergence Theorem 1, we already know that the primal error satisfies $h^{(k)} =$

$\mathrm{E}[f(\boldsymbol{x}^{(k)})] - f(\boldsymbol{x}^*) \le \frac{C}{k+2}$ in any iteration k, where we use the notation $C := 2C_f(1+\delta)$.

In the last third of the k iterations, we will now suppose that $g^{(k)}$ always stays larger than $\frac{\beta C}{K+2}$. We will derive a contradiction to this assumption. We write $D := K+2$ to simplify the notation. Formally, we assume that

$$g^{(k)} > \tfrac{\beta C}{D} \quad \text{for} \quad k \in \big\{ \lceil \mu D \rceil - 2 \ , \ \dots \ , \ K \big\} \ .$$

Here the parameter $0 < \mu < 1$ is arbitrary fixed, but we will see later that a good choice for this parameter is given by $\mu := \frac{2}{3}$.

Now employing the crucial improvement bound from Lemma 5 for the choice of $\gamma := \frac{2}{k+2}$, we have $h^{(k+1)} \le h^{(k)} - \gamma g^{(k)} + \frac{\gamma^2}{2} C_f(1+\delta)$. This bound from Lemma 5 holds no matter if $\boldsymbol{x}^{(k+1)}$ is obtained by using the pre-defined step-size, or using line-search, or by re-optimizing over the previous directions, since $f(\boldsymbol{x}^{(k+1)}_{\text{Re-Opt}}) \le f(\boldsymbol{x}^{(k+1)}_{\text{Line-Search}}) \le f(\boldsymbol{x}^{(k+1)}_{\gamma})$. Therefore, we have

$$\begin{aligned} h^{(k+1)} &\le h^{(k)} - \tfrac{2}{k+2} g^{(k)} + \tfrac{2}{(k+2)^2} C_f(1+\delta) \\ &= h^{(k)} - \tfrac{2}{k+2} g^{(k)} + \tfrac{C}{(k+2)^2} \ . \end{aligned}$$

Plugging in our assumption that the duality gap is still "large", we obtain

$$h^{(k+1)} < h^{(k)} - \tfrac{2}{k+2} \tfrac{\beta C}{D} + \tfrac{C}{(k+2)^2} \ .$$

Now we use that in our last third of the steps, our $\gamma := \frac{2}{k+2}$ is neither too large nor too small: More precisely, if we define $k_{\min} := \lceil \mu D \rceil - 2$ (note that $k_{\min} \ge 0$ if $K \ge \frac{1-\mu}{\mu} 2$), and consider the steps in $k_{\min} \le k \le K$, then $\mu D \le k+2 \le D$, so that our bound now reads as

$$\begin{aligned} h^{(k+1)} &< h^{(k)} - \tfrac{2}{D} \tfrac{\beta C}{D} + \tfrac{C}{(\mu D)^2} \\ &= h^{(k)} - \tfrac{2\beta C - C/\mu^2}{D^2} \ . \end{aligned}$$

We will now sum up this inequality over the last third of the steps from $k = k_{\min}$ up to $k = K$. These are at least $K - k_{\min} + 1 = K - (\lceil \mu D \rceil - 2) + 1 \ge (1-\mu) D =: n_3$ many steps, resulting in

$$\begin{aligned} h^{(K+1)} &< h^{(k_{\min})} - n_3 \tfrac{2\beta C - C/\mu^2}{D^2} \\ &\le \tfrac{C}{\mu D} - n_3 \tfrac{2\mu\beta - 1/\mu}{D} \tfrac{C}{\mu D} \\ &= \tfrac{C}{\mu D} \left(1 - n_3 \tfrac{2\mu\beta - 1/\mu}{D} \right) \ . \end{aligned}$$

here in the last inequality we have just used the primal convergence Theorem 1 giving $h^{(k_{\min})} \le \frac{C}{k_{\min}+2} \le \frac{C}{\mu D}$. This completes the proof, since we arrive at the contradiction that the primal error becomes negative, i.e. $h^{(K+1)} < 0$, when we plug in the claimed values for $\mu := \frac{2}{3}$ and $\beta := \frac{27}{8}$. Indeed, this pair of values will make the following term become zero: $1 - n_3 \frac{2\mu\beta - 1/\mu}{D} = 1 - (1-\mu)(2\mu\beta - 1/\mu) = 1 - \frac{1}{3}(2\frac{9}{4} - \frac{3}{2}) = 0$.

Therefore, our assumption on the gap is refuted, and we have proven the claimed bound. □

It is possible to obtain small duality gap within a slightly smaller number of iterations, corresponding to $\beta \approx 2$, if a constant step-size is used in the second half of the iterations, as formalized in the following theorem. The proof follows the idea of (Clarkson, 2010, Section 7).

Theorem 6 (Primal-Dual Convergence, Two-Regimes Variant)**.** *Suppose Algorithm 1, 2, 3 or 4 is run for $K \ge 1$ iterations, and then continued for another $K+1$ iterations, now with the fixed step-size $\gamma^{(k)} := \frac{2}{K+2}$ for all subsequent steps $K \le k \le 2K+1$.*

Then the algorithm has an iterate $\boldsymbol{x}^{(\hat{k})}$, $K \le \hat{k} \le 2K+1$, with duality gap bounded by

$$g(\boldsymbol{x}^{(\hat{k})}) \le \frac{2C_f}{K+2}(1+\delta) \ ,$$

where $\delta \ge 0$ is the accuracy to which the internal linear subproblems are solved.

Proof. Following the idea of (Clarkson, 2010, Section 7): By our previous Theorem 1 we already know that the primal error satisfies $h(\boldsymbol{x}^{(K)}) = f(\boldsymbol{x}^{(K)}) - f(\boldsymbol{x}^*) \le \frac{2C}{K+2}$ after K iterations, again using the notation $C := C_f(1+\delta)$.

In the subsequent $K+1$ iterations, we will now suppose that $g(\boldsymbol{x}^{(k)})$ always stays larger than $\frac{2C}{K+2}$. We will try to derive a contradiction to this assumption. Putting the assumption $g(\boldsymbol{x}^{(k)}) > \frac{2C}{K+2}$ into the step improvement bound given by Lemma 5, we get that

$$\begin{aligned} f(\boldsymbol{x}^{(k+1)}) - f(\boldsymbol{x}^{(k)}) &\le -\gamma^{(k)} g(\boldsymbol{x}^{(k)}) + \tfrac{{\gamma^{(k)}}^2}{2} C \\ &< -\gamma^{(k)} \tfrac{2C}{K+2} + \tfrac{{\gamma^{(k)}}^2}{2} C \end{aligned}$$

holds for any step size $\gamma^{(k)} \in (0,1]$. Now using the fixed step-size $\gamma^{(k)} = \frac{2}{K+2}$ in the iterations $k \ge K$ of the algorithm, this reads as

$$\begin{aligned} f(\boldsymbol{x}^{(k+1)}) - f(\boldsymbol{x}^{(k)}) &< -\tfrac{2}{K+2} \tfrac{2C}{K+2} + \tfrac{2}{(K+2)^2} C \\ &= -\tfrac{2C}{(K+2)^2} \end{aligned}$$

Summing up over the additional steps, we obtain

$$\begin{aligned} f(\boldsymbol{x}^{(2K+2)}) - f(\boldsymbol{x}^{(K)}) &= \sum_{k=K}^{2K+1} f(\boldsymbol{x}^{(k+1)}) - f(\boldsymbol{x}^{(k)}) \\ &< -(K+2) \tfrac{2C}{(K+2)^2} = -\tfrac{2C}{K+2} \ , \end{aligned}$$

which together with our known primal approximation error $f(\boldsymbol{x}^{(K)}) - f(\boldsymbol{x}^*) \le \frac{2C}{K+2}$ would result in $f(\boldsymbol{x}^{(2K+2)}) - f(\boldsymbol{x}^*) < 0$, a contradiction. Therefore there must exist $\hat{k}$, $K \le \hat{k} \le 2K+1$, with $g(\boldsymbol{x}^{(\hat{k})}) \le \frac{2C}{K+2}$. □

C. Optimality of the Trade-Off between Sparsity and Approximation Quality

Lemma' 3. *For $f(\boldsymbol{x}) := \|\boldsymbol{x}\|_2^2$, and $1 \le k \le n$, it holds that* $\min_{\substack{\boldsymbol{x}\in\Delta_n \\ \operatorname{card}(\boldsymbol{x})\le k}} f(\boldsymbol{x}) = \frac{1}{k}$.

Proof. We prove the inequality $\min_{\boldsymbol{x}..} f(\boldsymbol{x}) \ge \frac{1}{k}$ by induction on k. The base-case $k = 1$ follows since $f(\boldsymbol{x}) = \|\boldsymbol{x}\|_2 = \|\boldsymbol{x}\|_1 = 1$ for any unit length vector $\boldsymbol{x} \in \Delta_n$ having just a single non-zero entry. For $k > 1$, we use that for every $\boldsymbol{x} \in \Delta_n$ of sparsity $\operatorname{card}(\boldsymbol{x}) \le k$, we can pick a coordinate i with $x_i \ne 0$, and write $\boldsymbol{x} = (1-\gamma)\boldsymbol{v} + \gamma\mathbf{e}_i$ as the sum of two orthogonal vectors: $\boldsymbol{v}$ and a unit basis vector $\mathbf{e}_i$, where $\boldsymbol{v} \in \Delta_n$ of sparsity $\le k-1$, $v_i = 0$, and $\gamma = x_i$. Therefore

$$\begin{array}{rcl} f(\boldsymbol{x}) = \|\boldsymbol{x}\|_2^2 & = & ((1-\gamma)\boldsymbol{v} + \gamma\mathbf{e}_i)^T((1-\gamma)\boldsymbol{v} + \gamma\mathbf{e}_i) \\ & = & (1-\gamma)^2\boldsymbol{v}^T\boldsymbol{v} + \gamma^2 \\ & \ge & (1-\gamma)^2\frac{1}{k-1} + \gamma^2 \\ & \ge & \min\limits_{0\le\beta\le1}(1-\beta)^2\frac{1}{k-1} + \beta^2 = \frac{1}{k}. \end{array}$$

In the first inequality we have applied the induction hypothesis for $\boldsymbol{v} \in \Delta_n$ of sparsity $\le k-1$.
Equality: The value $f(\boldsymbol{x}) = \frac{1}{k}$ is attained by setting k of the coordinates of $\boldsymbol{x}$ to $\frac{1}{k}$ each. □

The lower bound here also extends to prove that the obtained duality gap $g(\boldsymbol{x})$ is best possible:

Lemma' 4. *For $f(\boldsymbol{x}) := \|\boldsymbol{x}\|_2^2$, and any $k \in \mathbb{N}$, $k < n$, it holds that* $g(\boldsymbol{x}) \ge \frac{2}{k}$ $\quad\forall \boldsymbol{x} \in \Delta_n$ *s.t.* $\operatorname{card}(\boldsymbol{x}) \le k$.

Proof. $g(\boldsymbol{x}) = \boldsymbol{x}^T\nabla f(\boldsymbol{x}) - \min_i(\nabla f(\boldsymbol{x}))_i = 2(\boldsymbol{x}^T\boldsymbol{x} - \min_i x_i)$. We now use $\min_i x_i = 0$ because $\operatorname{card}(\boldsymbol{x}) < n$, and that by Lemma 3 we have $\boldsymbol{x}^T\boldsymbol{x} = f(\boldsymbol{x}) \ge \frac{1}{k}$. □

D. Relating Curvature to Lipschitz-Continuous Gradient

For any choice of norm $\|.\|$, the curvature constant C_f can be upper bounded as follows:

Lemma 7. *Let f be a convex and differentiable function with its gradient ∇f is Lipschitz-continuous w.r.t. some norm $\|.\|$ over the domain $\mathcal{D}$ with Lipschitz-constant $L > 0$. Then*

$$C_f \le \operatorname{diam}_{\|.\|}(\mathcal{D})^2 L \ .$$

Proof. By (Nesterov, 2004, Lemma 1.2.3), we have that for any $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}$,

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) - \langle \boldsymbol{y} - \boldsymbol{x}, \nabla f(\boldsymbol{x})\rangle \le \tfrac{L}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2$$

We want to use this upper bound in the definition (3) of the curvature constant. Observing that for any $\boldsymbol{x}, \boldsymbol{s} \in \mathcal{D}$, we have that also $\boldsymbol{y} := \boldsymbol{x} + \gamma(\boldsymbol{s} - \boldsymbol{x}) \in \mathcal{D}$ and $\frac{1}{\gamma^2}\|\boldsymbol{y} - \boldsymbol{x}\|^2 = \|\boldsymbol{s} - \boldsymbol{x}\|^2$, we can therefore upper bound the curvature as

$$\begin{array}{rcl} C_f \le \sup \frac{2}{\gamma^2}\frac{L}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2 = & & \sup L\|\boldsymbol{s} - \boldsymbol{x}\|^2 \\ & \le & L\operatorname{diam}_{\|.\|}(\mathcal{D})^2 \ , \end{array}$$

which is the claimed bound. □

E. Approximating the Top Eigenvalue of a Matrix

For optimization over bounded trace-norm (see Section 4.3), we have seen that the linear subproblem to be solved in every iteration of the Frank-Wolfe algorithm amounts to finding an approximate top eigenvalue (or singular vector pair). The running time of the standard Lanczos' algorithm for this subproblem is bounded as follows:

Proposition 8 (Kuczyński & Woźniakowski (1992))**.** *For any matrix $M \in \mathbb{R}^{m\times n}$, and $\varepsilon' > 0$, Lanczos' algorithm returns a pair of unit vectors $(\boldsymbol{u}, \boldsymbol{v})$ s.t. $\boldsymbol{u}^TM\boldsymbol{v} \ge \sigma_1(M) - \varepsilon'$, with high probability, using at most $O\Big(N_M\frac{\log(n+m)\sqrt{L}}{\sqrt{\varepsilon'}}\Big)$ arithmetic operations.*

Here N_M is the number of non-zero entries of the input matrix M, and L is an upper bound on $\sigma_1(M)$. Note that for the Frank-Wolfe Algorithms 2, 3, and 4, the subproblem accuracy needs to be chosen not larger than $\varepsilon' := \frac{\delta C_f}{k+2}$ in iteration k, which is in $O(\varepsilon)$.

Compared to SVD taking at least cubic time in $n+m$, such an approximate computation of only one approximate eigenvector (or singular vector pair) is much more efficient, see also (Jaggi & Sulovský, 2010).

Randomized Subproblems. Note that in general, if the linear subproblem in each step is solved approximately only *in expectation*, then the step-improvement bound from Lemma 5 still holds in expectation (conditioned on the previous iterate). Therefore, the primal as well as primal-dual convergence bounds (from the main Theorems 1 and 2) do still hold in expectation in this case.