Linearly Convergent Frank-Wolfe with Backtracking Line-Search

Fabian Pedregosa Google Research
Geoffrey Négiar University of California, Berkeley
Armin Askari EPFL, Lausanne
Martin Jaggi EPFL, Lausanne

Abstract

Structured constraints in Machine Learning have recently brought the Frank-Wolfe (FW) family of algorithms back into the spotlight. While the classical FW algorithm has poor local convergence properties, Away-steps FW and Pairwise FW have emerged as improved variants with faster convergence. However, these improved variants suffer from two practical limitations: they require at each iteration to solve a 1-dimensional minimization problem to set the step-size and also require the Frank-Wolfe linear subproblems to be solved exactly. In this paper, we propose variants of Away-steps and Pairwise FW that lift both restrictions simultaneously. The proposed methods set the step-size based on a sufficient decrease condition, and do not require prior knowledge of the objective. Furthermore, they inherit all the favorable convergence properties of the exact line-search version, including linear convergence for strongly convex functions over polytopes. Benchmarks on different machine learning problems illustrate large performance gains of the proposed variants.

1 Introduction

The Frank-Wolfe (FW) or conditional gradient algorithm (Frank and Wolfe, 1956; Levitin and Polyak, 1966; Demyanov and Rubinov, 1967) is a method for constrained optimization that solves problems of the form

$$\underset{\boldsymbol{x}\in\operatorname{conv}(\mathcal{A})}{\text{minimize}}\ f(\boldsymbol{x})\,, \tag{OPT}$$

where f is a smooth function for which we have access to its gradient and $\operatorname{conv}(\mathcal{A})$ is the convex hull of $\mathcal{A}$; a bounded but potentially infinite set of elements in $\mathbb{R}^p$ which we will refer to as *atoms*.

The FW algorithm is one of the oldest methods for non-linear constrained optimization and has experienced a renewed interest in recent years due to its applications in machine learning and signal processing (Jaggi, 2013). Despite some favorable properties, the local convergence of the FW algorithm is known to be slow, achieving only a sublinear rate of convergence for strongly convex functions when the solution lies in the boundary (Canon and Cullum, 1968). To overcome these limitations, variants of the FW algorithms with better convergence properties have been proposed. Two of these variants, the Away-steps FW (Guélat and Marcotte, 1986) and Pairwise FW (Lacoste-Julien and Jaggi, 2015) enjoy a linear rate of convergence over polytopes (Lacoste-Julien and Jaggi, 2015).

Despite this theoretical breakthrough, Away-steps and Pairwise FW are not yet practical off-the-shelf solvers due to two main limitations. The first and most important is that both variants rely on an *exact line-search*. That is, they require to solve at each iteration 1-dimensional subproblems of the form

$$\underset{\gamma\in[0,\gamma_{\max}]}{\arg\min}\ f(\boldsymbol{x}_t + \gamma \boldsymbol{d}_t)\ , \tag{1}$$

where $\boldsymbol{d}_t$ is the update direction and $\gamma_{\max}$ is the maximum admissible step-size. In a few cases like quadratic objectives, the exact line-search subproblem has a closed form solution. In most other cases, it is a costly optimization problem that needs to be solved at each iteration, making these methods impractical. The second limitation is that they require access to an *exact Linear Minimization Oracle* (LMO), which leaves out important cases like minimization over a trace norm ball where the LMO is computed up to some predefined tolerance. In this paper we develop methods that lift both limitations simultaneously.

Our **main contribution** is the design and analysis of variants of Away-steps and Pairwise FW that *i*) don't

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

	Related work	non-convex analysis	approximate subproblems	linear convergence	adaptive step-size	bounded backtracking
Frank-Wolfe	*This work*	✓	✓	✓	✓	✓
	(Lacoste-Julien and Jaggi, 2015)	✗	✗	✓	✗	N/A
	(Beck et al., 2015)	✗	✓†	✗	✓	✗
	(Dunn, 1980)	✓	✗	✗	✓	✗
MP	*This work*	✓	✓	✓	✓	✓
	(Locatello et al., 2017)	✗	✓	✓	✗	N/A

Table 1: **Comparison with related work.** *non-convex analysis*: convergence guarantees for problems with a non-convex objective. *approximate subproblems*: convergence guarantees cover the case in which linear subproblems are solved approximately. *linear convergence*: guaranteed linear rate of convergence (under hypothesis). *adaptive step-size*: step-size is set using local information of the objective. *bounded backtracking*: explicit bound for the total number of inner iterations in adaptive step-size methods. † = assumes cartesian product domain.

require access to an exact line-search or knowledge of properties of the objective like its curvature or Lipschitz constant, and *ii*) admits the FW subproblems to be solved approximately. We describe our approach in §2. Although our main motivation is to develop practical variants of Away-steps and Pairwise FW, we also show that this technique extends to other methods like FW and Matching Pursuit.

We develop in §3 a convergence rate analysis for the proposed methods. The obtained rates match asymptotically the best known bounds on convex, strongly convex and non-convex problems, including linear convergence for strongly convex functions.

Finally, we show in §4 benchmarks between the proposed and related methods, and discuss the importance of large step-sizes in Pairwise FW.

1.1 Related work

We comment on the most closely related ideas, summarized in Table 1.

Away-Steps FW (Guélat and Marcotte, 1986) is a popular variant of FW that adds the option to move away from an atom in the current representation of the iterate. In the case of a polytope domain, it was recently shown to enjoy a linear convergence rate for strongly convex objectives (Garber and Hazan, 2013; Lacoste-Julien and Jaggi, 2015). Pairwise FW (Lacoste-Julien and Jaggi, 2015) simplifies the above-described variant by replacing the two kinds of steps by a single step modifying the weights of only two atoms. It generalizes the algorithm of Mitchell et al. (1974) used in geometry and SMO (Platt, 1998) for training SVMs. These methods all require exact line-search.

Variants of FW that, like the proposed methods, set the step-size based on a local decrease condition have been described by Dunn (1980) and Beck et al. (2015), but none of these methods achieve a linear convergence rate to the best of our knowledge.

Matching Pursuit (MP) (Mallat and Zhang, 1993) is an algorithm for constrained optimization problems of the form (OPT) with $\text{conv}(\mathcal{A})$ replaced by $\text{linspan}(\mathcal{A})$, the linear span of $\mathcal{A}$. Locatello et al. (2018) has recently shown that MP and FW are deeply related. We show that our algorithm and convergence results also extend naturally to MP, and as a byproduct of our analysis we obtain the first convergence rate for MP on non-convex objectives to the best of our knowledge.

Notation. Throughout the paper we denote vectors and vector-valued functions in lowercase boldface (e.g. $\boldsymbol{x}$ or $\mathbf{arg\,min}$), matrices in uppercase boldface letters (e.g. $\boldsymbol{D}$), and sets in caligraphic letters (e.g., $\mathcal{A}$). We say a function f is L-smooth if it is differentiable and its gradient is L-Lipschitz continuous, that is, if it verifies $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|$ for all $\boldsymbol{x}, \boldsymbol{y}$ in the domain. A function is μ-strongly convex if $f - \frac{\mu}{2}\|\cdot\|^2$ is convex. $\|\cdot\|$ denotes the euclidean norm.

2 Methods

In this section we describe the core part of our contribution, which is a strategy to select the step-size in FW-type algorithms.

Since this strategy can be applied very broadly to Frank-Wolfe variants including Away-steps, Pairwise, classical FW and Matching Pursuit, we describe it within the context of a generic FW-like algorithm. This generic algorithm is detailed in Algorithm 1 and depends on two key functions: `update_direction` and `step_size`. The first one computes the direction that we will follow to compute the next iterate and its implementation will depend on the FW variant. The second one will choose an appropriate step-size based upon local information of the objective and is the key novelty of this algorithm. We now describe them in more detail.

Algorithm 1: Generic FW with backtracking

1 **Input:** $\boldsymbol{x}_0 \in \text{conv}(\mathcal{A})$, initial Lipschitz estimate $L_{-1} > 0$, tolerance $\varepsilon \geq 0$, subproblem quality $\delta \in (0, 1]$
2 **for** $t = 0, 1 \dots$ **do**
3 $\quad \boldsymbol{d}_t, \gamma_t^{\max} = \text{update\_direction}(\boldsymbol{x}_t, \nabla f_t)$
4 $\quad g_t = \langle -\nabla f(\boldsymbol{x}_t), \boldsymbol{d}_t \rangle$
5 $\quad$ **if** $g_t \leq \delta\varepsilon$ **then return** $\boldsymbol{x}_t$;
6 $\quad \gamma_t, L_t = \text{step\_size}(f, \boldsymbol{d}_t, \boldsymbol{x}_t, g_t, L_{t-1}, \gamma_t^{\max})$
7 $\quad \boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \gamma_t \boldsymbol{d}_t$

Algorithm 2: Backtracking for FW variants

1 **Procedure** `step_size`(f,$\boldsymbol{d}_t$, $\boldsymbol{x}_t$, g_t, L_{t-1}, $\gamma_{\max}$)
2 $\quad$ Choose $\tau > 1$, $\eta \leq 1$
3 $\quad$ Choose $M \in [\eta L_{t-1}, L_{t-1}]$
4 $\quad \gamma = \min\left\{g_t/(M\|\boldsymbol{d}_t\|^2), \gamma_{\max}\right\}$
5 $\quad$ **while** $f(\boldsymbol{x}_t + \gamma \boldsymbol{d}_t) > Q_t(\gamma, M)$ **do**
6 $\quad\quad M = \tau M$
7 $\quad\quad \gamma = \min\left\{g_t/(M\|\boldsymbol{d}_t\|^2), \gamma_{\max}\right\}$
8 $\quad$ **end**
9 $\quad$ **return** γ, M

2.1 Update direction

In this subsection we describe `update_direction` in Algorithm 1, the function that computes the update direction $\boldsymbol{d}_t$ and the maximum allowable step-size $\gamma_t^{\max}$. While its implementation varies according to the FW variant, all of them require to solve one or two linear problems, often referred to as linear minimization oracle (LMO).

The first of these subproblems is the same for all variants and consists in finding $\boldsymbol{s}_t$ in the domain such that:

$$\langle \nabla f(\boldsymbol{x}_t), \boldsymbol{s}_t - \boldsymbol{x}_t \rangle \leq \delta \min_{\boldsymbol{s} \in \mathcal{A}} \langle \nabla f(\boldsymbol{x}_t), \boldsymbol{s} - \boldsymbol{x}_t \rangle \ . \tag{2}$$

Here, we introduce a *subproblem quality* parameter $\delta \in (0, 1]$ that allows this subproblem to be solved approximately. When $\delta = 1$, the problem is solved exactly and becomes $\arg\min_{\boldsymbol{s} \in \mathcal{A}} \langle \nabla f(\boldsymbol{x}_t), \boldsymbol{s} \rangle$, which consists in selecting the atom that correlates the most with the steepest descent direction, $-\nabla f(\boldsymbol{x}_t)$.

Away-steps and Pairwise FW will also require to solve another linear subproblem, this time over the *active set* $\mathcal{S}_t$. This is the set of atoms with non-zero weight in the decomposition of $\boldsymbol{x}_t$. More formally, the active set $\mathcal{S}_t \subseteq \mathcal{A}$ is the set of atoms that have non-zero weight $\boldsymbol{\alpha}_{\boldsymbol{s},t} > 0$ in the expansion $\boldsymbol{x}_t = \sum_{\boldsymbol{s} \in \mathcal{S}_t} \boldsymbol{\alpha}_{\boldsymbol{v},t} \boldsymbol{s}$.

The linear subproblem that needs to be solved consists in finding $\boldsymbol{v}_t$ such that:

$$\langle \nabla f(\boldsymbol{x}_t), \boldsymbol{x}_t - \boldsymbol{v}_t \rangle \leq \delta \min_{\boldsymbol{v} \in S_t} \langle \nabla f(\boldsymbol{x}_t), \boldsymbol{x}_t - \boldsymbol{v} \rangle \ . \tag{3}$$

Unlike the previous linear subproblem, this time the problem is over the typically much smaller active set $\mathcal{S}_t$. As before, $\delta \in (0, 1]$ allows this subproblem to be solved approximately. When $\delta = 1$, the subproblem becomes $\arg\max_{\boldsymbol{v} \in \mathcal{S}_t} \langle \nabla f(\boldsymbol{x}_t), \boldsymbol{v} \rangle$, which can be interpreted as selecting the atom in the active set that correlates the most with the steepest ascent direction $\nabla f(\boldsymbol{x}_t)$.

FW, AFW and PFW then combine the solution to these linear subproblems in different ways, and Line 3's `update_direction` is implemented as:

- FW returns $\boldsymbol{d}_t = \boldsymbol{s}_t - \boldsymbol{x}_t$ and $\gamma_t^{\max} = 1$: the next iterate will be a convex combination of $\boldsymbol{x}_t$ and $\boldsymbol{s}_t$.
- AFW considers directions $\boldsymbol{s}_t - \boldsymbol{x}_t$ and $\boldsymbol{x}_t - \boldsymbol{v}_t$, and chooses the one that correlates the most with $-\nabla f(\boldsymbol{x}_t)$. $\gamma_t^{\max} = 1$ if $\boldsymbol{d}_t = \boldsymbol{s}_t - \boldsymbol{x}_t$ and $\alpha_{\boldsymbol{v}_t}/(1-\alpha_{\boldsymbol{v}_t})$ otherwise, where $\alpha_{\boldsymbol{v}_t}$ is the weight associated with $\boldsymbol{v}_t$ in the decomposition of $\boldsymbol{x}_t$ as a convex combination of atoms.
- PFW uses $\boldsymbol{d}_t = \boldsymbol{s}_t - \boldsymbol{v}_t$, shifting weight from $\boldsymbol{v}_t$ to $\boldsymbol{s}_t$ in our current iterate, and $\gamma_t^{\max} = \alpha_{\boldsymbol{v}_t}$.
- MP uses $\boldsymbol{d}_t = \boldsymbol{s}_t$ and $\gamma^{\max} = +\infty$, since the constraint set is not bounded.

2.2 Backtracking line-search

In this subsection we describe the step-size selection routine `step_size` (Algorithm 2). This is the main novelty in the proposed algorithms, and allows for the step-size to be computed using only *local* properties of the objective, as opposed to other approaches that use global quantities like the gradient's Lipschitz constant. As we will see in §4, this results in step-sizes that are often more than an order of magnitude larger than those estimated using global quantities.

Minimizing the exact line-search objective $\gamma \mapsto f(\boldsymbol{x}_t + \gamma \boldsymbol{d}_t)$ yields the highest decrease in objective but can be a costly optimization problem. To overcome this, we will replace the exact line-search objective with the following quadratic approximation:

$$Q_t(\gamma, M) = f(\boldsymbol{x}_t) - \gamma g_t + \frac{\gamma^2 M}{2} \|\boldsymbol{d}_t\|^2 \ . \tag{4}$$

This approximation has the advantage that its minimum over $\gamma \in [0, \gamma^{\max}]$ can be computed in closed form, which gives the step-size used in line 4:

$$\gamma_M^\star = \min\left\{ \frac{g_t}{M\|\boldsymbol{d}_t\|^2}, \gamma^{\max} \right\} \ , \tag{5}$$

The quality of this quadratic approximation will depend on the *Lipschitz estimate* parameter M. This parameter needs to be carefully selected to maintain the convergence guarantees of exact line-search, while keeping the number of objective function evaluations to a minimum.

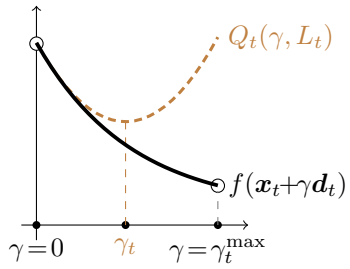
This is achieved through the strategy implemented in Algorithm 2. The algorithm initializes the Lipschitz estimate M to a value between ηL_{t-1} and the previous iterate L_{t-1}, where η is a user-defined parameter (default values discussed later). A value of $\eta = 1$ is admissible but would not allow the Lipschitz estimate to decrease through the optimization, and we have observed empirically a drastic benefit in doing so.

The algorithm then defines a candidate step-size γ (Line 4) and checks whether the following *sufficient decrease* condition is verified for this step-size

$$f(\boldsymbol{x}_t+\gamma\boldsymbol{d}_t) \leq Q_t(\gamma, M)\ , \gamma = \min\left\{g_t/(M\|\boldsymbol{d}_t\|^2), \gamma^{\max}\right\}.$$

If it is not verified, we increase this constant by a power factor of $\tau > 1$ (Line 6). By the properties of L-smooth functions, we know that this condition is verified for all $M \geq L$, and so this loop has a finite termination.

Once this condition is verified, the current step-size is accepted and the value of M is assigned the name L_t. Geometrically, the sufficient decrease condition ensures that the quadratic approximation is an upper bound at its constrained minimum of the line-search objective. We emphasize that this upper bound does not need to be a global one, as it only holds at γ_t. This allows for smaller L_t than the global Lipschitz constant L, and therefore larger step-sizes. As we will see in §3, this translates into faster convergence rates that depend on L_t, as well as faster empirical convergence (§4).



Default and initial parameters. Algorithm 1 requires an (arbitrary) initial value for the Lipschitz estimate L_{-1}. We found the following heuristic using the definition of Lipschitz continuity of the gradient to work well in practice. Select a small constant ε, say 10^{-3}, and compute an initial value as $L_{-1} = \|\nabla f(\boldsymbol{x}_0) - \nabla f(\boldsymbol{x}_0 + \varepsilon\boldsymbol{d}_0)\|/(\varepsilon\|\boldsymbol{d}_t\|)$.

The `step_size` depends on hyperparameters η and τ. Although the algorithm is guaranteed to converge for any $\eta \leq 1$, $\tau > 1$, we recommend $\eta = 0.9, \tau = 2$, as we found that it performs well in a variety of scenarios. These are the values used throughout benchmarks §4.

This method also requires to choose the initial value of the Lipschitz estimate M to a value between ηL_{t-1} and L_{t-1}. A choice that we found to work remarkably well in practice is to initialize it to

$$M = \text{clip}_{[\eta L_{t-1}, L_{t-1}]}\left(\frac{g_t^2}{2(f_{t-1} - f_t)\|\boldsymbol{d}_t\|^2}\right). \quad (6)$$

The value inside the clip function corresponds to the optimal value of M for a quadratic interpolation between the previous two iterates and the derivative of the line-search objective $f(\boldsymbol{x}_t + \gamma\boldsymbol{d}_t)$ at $\gamma = 0$. Since this value might be outside of the interval $[\eta L_{t-1}, L_{t-1}]$, we clip the result to this interval.

Pseudocode and implementation details. A practical implementation of these algorithms depends on other details that are not specific to the backtracking variant, such as efficiently maintaining the active-set in the case of Away-steps and Pairwise. For completeness, Appendix A contains a full pseudocode for all these algorithms. A Python implementation of these methods, as well as the benchmarks used in §4 will be made open source upon publication of this manuscript.

3 Analysis

In this section, we provide a convergence rate analysis of the proposed methods, showing that all enjoy a $\mathcal{O}(1/\sqrt{t})$ convergence rate for non-convex objectives (Theorem 2), a stronger $\mathcal{O}(1/t)$ convergence rate for convex objectives (Theorem 3), and for some variants linear convergence for strongly convex objectives over polytopes (Theorem 4).

Notation. In this section we make use of the following extra notation:

- For convenience we will refer to the variants of FW, Away-steps FW, Pairwise FW and MP with backtracking line-search as AdaFW, AdaAFW, AdaPFW and AdaMP respectively.
- We denote the *objective suboptimality* at step t as $h_t = f(\boldsymbol{x}_t) - \min_{\boldsymbol{x}\in\text{conv}(\mathcal{A})} f(\boldsymbol{x})$.
- *Good and bad steps.* Our analysis, as that of Lacoste-Julien and Jaggi (2015), relies on a notion of "good" and "bad" steps. We define bad steps as those that verify $\gamma_t = \gamma_t^{\max}$ and $\gamma_t^{\max} < 1$ and good steps as any step that is not a bad step. The name "bad steps" makes reference to the fact that we won't be able to bound non-trivially the improvement for these steps. For these steps we will only be able to guarantee that the objective is non-increasing. AdaAFW and AdaPFW both may have

Algorithm	Non-convex	Convex	Strongly convex
AdaAFW	$\mathcal{O}(\frac{1}{\delta\sqrt{t}})$	$\mathcal{O}(\frac{1}{\delta^2 t})$	$\mathcal{O}((1-\delta^2\rho)^t)$
AdaPFW	$\mathcal{O}(\frac{1}{\delta\sqrt{t}})$	$\mathcal{O}(\frac{1}{\delta^2 t})$	$\mathcal{O}((1-\delta^2\rho)^t)$
AdaFW	$\mathcal{O}(\frac{1}{\delta\sqrt{t}})$	$\mathcal{O}(\frac{1}{\delta^2 t})$	$\mathcal{O}(\frac{1}{\delta^2 t})$
AdaMP	$\mathcal{O}(\frac{1}{\delta\sqrt{t}})$	$\mathcal{O}(\frac{1}{\delta^2 t})$	$\mathcal{O}((1-\delta^2\rho_{\text{MP}})^t)$

Table 2: **Convergence rate summary** on non-convex, convex and strongly convex objectives. For non-convex objectives, bound is on the minimum FW gap (MP gap in the case of AdaMP), in other cases its on the objective suboptimality.

bad steps. Let us denote by N_t the number of "good steps" up to iteration t. We can lower bound the number of good steps by

$$N_t \geq t/2 \text{ for AdaAFW}, \tag{7}$$
$$N_t \geq t/(3|\mathcal{A}|! + 1) \text{ for AdaPFW} \tag{8}$$

where it is worth noting that the last bound for AdaPFW requires the set of atoms $\mathcal{A}$ to be finite. The proof of these bounds can be found in Appendix C.1 and are a direct translation of those in (Lacoste-Julien and Jaggi, 2015). We have found these bounds to be very loose, as in practice the fraction of bad/good steps is negligible, commonly of the order of 10^{-5} (see last column of the table in Figure 1).

- *Average and maximum of Lipschitz estimates.* In order to highlight the better convergence rates that can be obtained by adaptive methods we introduce the average and maximum estimate over good step-sizes. Let $\mathcal{G}_t$ denote the indices of good steps up to iteration t. Then we define the average and maximum Lipschitz estimate as

$$\overline{L}_t \stackrel{\text{def}}{=} \frac{1}{N_t}\sum_{k\in\mathcal{G}_t} L_k \tag{9}$$
$$L_t^{\max} \stackrel{\text{def}}{=} \max_{k\in\mathcal{G}_t} L_k \tag{10}$$

respectively. In the worst case, both quantities can be upper bounded by $\max\{\tau L, L_{-1}\}$ (Proposition 2), which can be used to obtain asymptotic convergence rates. This bound is however very pessimistic. We have found that in practice $\overline{L}_t$ is often more than 100 times smaller than L (see second to last column of the table in Figure 1).

Our new convergence rates are presented in the following theorems, which consider the cases of non-convex, convex and strongly convex objectives. The results are discussed in §3.5 and the proofs can be found in Appendix D, Appendix E and Appendix F respectively.

3.1 Overhead of backtracking

Evaluation of the sufficient decrease condition Algorithm 2 requires two extra evaluations of the objective function. If the condition is verified, then it is only evaluated at the current and next iterate. FW requires anyway to compute the gradient at these iterates, hence in cases in which the objective function is available as a byproduct of the gradient this overhead becomes negligible.

Furthermore, we can provide a bound on the total number of evaluations of the sufficient decrease condition:

Theorem 1. *Let n_t be the total number of evaluations of the sufficient decrease condition up to iteration t. Then we have*

$$n_t \leq \left[1 - \frac{\log\eta}{\log\tau}\right](t+1) + \frac{1}{\log\tau}\max\left\{\log\frac{\tau L}{L_{-1}}, 0\right\},$$

This result highlights the trade-off faced when choosing η. Minimizing it with respect to η gives $\eta = 1$, in which case $(1 - \log\eta/\log\tau) = 1$ and so there's an asymptotically vanishing number of failures in the sufficient decrease condition. Unfortunately, $\eta = 1$ also forbids the Lipschitz estimate to decrease along the optimization. Ideally, we would like η small enough so that the Lipschitz estimate decreases when it can, but not too small so that we waste too much time in failed sufficient decrease evaluations.

As mentioned before, we recommend parameters $\eta = 0.9$, $\tau = 2$. With these values, we have that $\left[1 - \frac{\log\eta}{\log\tau}\right] \leq 1.16$, and so asymptotically no more than 16% of the iterates will result in more than one evaluations of the sufficient decrease condition.

3.2 Non-convex objectives

Gap function. Convergence rates for convex and strongly convex functions are given in terms of the objective function suboptimality or a primal-dual gap. As the gap upper-bounds (i.e. certifies) the suboptimality, the latter is a stronger result in this scenario. In the case of non-convex objectives, as is common for first order methods, we will only be able to guarantee convergence to a stationary point, defined as any element $\boldsymbol{x}^\star \in \text{conv}(\mathcal{A})$ such that $\langle\nabla f(\boldsymbol{x}^\star), \boldsymbol{x} - \boldsymbol{x}^\star\rangle \geq 0$ for all $\boldsymbol{x} \in \text{conv}(\mathcal{A})$ (Bertsekas, 1999).

Following Lacoste-Julien (2016); Reddi et al. (2016), for FW variants we will use as convergence

criterion the FW gap, defined as $g^{\text{FW}}(\boldsymbol{x}) = \max_{\boldsymbol{s}\in\text{conv}(\mathcal{A})}\langle\nabla f(\boldsymbol{x}), \boldsymbol{x}-\boldsymbol{s}\rangle$. From the definition of stationary point it is clear that the FW gap is zero only at a stationary point. These rates are also valid for AdaMP, albeit for the more appropriate gap function g^{MP} detailed in Appendix D.

Theorem 2. *Let $\boldsymbol{x}_t$ denote the iterate generated by any of the proposed algorithms after t iterations, with $N_{t+1} \geq 1$. Then we have:*

$$\lim_{t\to\infty} g(\boldsymbol{x}_t) = 0 \qquad \textit{and} \tag{11}$$

$$\min_{k=0,\dots,t} g(\boldsymbol{x}_k) \leq \frac{C_t}{\delta\sqrt{N_{t+1}}} = \mathcal{O}\left(\frac{1}{\delta\sqrt{t}}\right), \tag{12}$$

where $C_t = \max\{2h_0, L_t^{\max}\operatorname{diam}(\mathcal{A})^2\}$ and $g = g^{FW}$ is the FW gap for AdaFW, AdaAFW, AdaPFW and $C_t = \operatorname{radius}(\mathcal{A})\sqrt{2h_0\overline{L}_{t+1}}$ and $g = g^{MP}$ is the MP gap for AdaMP.

3.3 Convex objectives

For convex objectives we will be able to improve the results of Theorem 2. We will first state the convergence results for FW variants and then for MP.

For adaptive FW variants, we will be able to give an $\mathcal{O}(1/\delta^2 t)$ convergence rate on the primal-dual gap, which trivially implies a bound on the objective suboptimality. In order to define the primal-dual gap, we define the following *dual* objective function

$$\psi(\boldsymbol{u}) \stackrel{\text{def}}{=} -f^*(\boldsymbol{u}) - \sigma_{\text{conv}(\mathcal{A})}(-\boldsymbol{u}) \;, \tag{13}$$

where f^* denotes the convex conjugate of f and $\sigma_{\text{conv}(\mathcal{A})}(\boldsymbol{x}) \stackrel{\text{def}}{=} \sup\{\langle\boldsymbol{x},\boldsymbol{a}\rangle : \boldsymbol{a}\in\text{conv}(\mathcal{A})\}$ is the support function over $\text{conv}(\mathcal{A})$, which is the convex conjugate of the indicator function. Note that ψ is concave and that when f convex, we have by duality $\min_{\boldsymbol{x}\in\text{conv}(\mathcal{A})} f(\boldsymbol{x}_t) = \max_{\boldsymbol{u}\in\mathbb{R}^p}\psi(\boldsymbol{u})$.

Theorem 3 (FW variants)**.** *Let f be convex, $\boldsymbol{x}_t$ denote the iterate generated by any of the proposed FW variants (AdaFW, AdaAFW, AdaPFW) after t iterations, with $N_t \geq 1$, and let $\boldsymbol{u}_t$ be defined recursively as $\boldsymbol{u}_0 = \nabla f(\boldsymbol{x}_0)$, $\boldsymbol{u}_{t+1} = (1-\xi_t)\boldsymbol{u}_t + \xi_t\nabla f(\boldsymbol{x}_t)$, where $\xi_t = 2/(\delta N_t + 2)$ if t is a good step and $\xi_t = 0$ otherwise. Then we have:*

$$h_t \leq f(\boldsymbol{x}_t) - \psi(\boldsymbol{u}_t) \tag{14}$$

$$\leq \frac{2\overline{L}_t\operatorname{diam}(\mathcal{A})^2}{\delta^2 N_t + \delta} + \frac{2(1-\delta)}{\delta^2 N_t^2 + \delta N_t}\big(f(\boldsymbol{x}_0) - \psi(\boldsymbol{u}_0)\big)$$

$$= \mathcal{O}\left(\frac{1}{\delta^2 t}\right). \tag{15}$$

3.4 Strongly convex objectives

The next result states the linear convergence of some algorithm variants and uses the notions of pyramidal width (PWidth) and minimal directional width (mDW) that have been developed in (Lacoste-Julien, 2016) and (Locatello et al., 2017) respectively, which we state in Appendix B for completeness. We note that the pyramidal width of a set $\mathcal{A}$ is lower bounded by the minimal width over all subsets of atoms, and thus is strictly greater than zero if the number of atoms is finite. The minimal directional width is a much simpler quantity and always strictly greater than zero by the symmetry of our domain.

Theorem 4 (Linear convergence rate for strongly convex objectives)**.** *Let f be μ–strongly convex. Then for AdaAFW, AdaPFW or AdaMP we have the following linear decrease for each good step t:*

$$h_{t+1} \leq (1-\delta^2\rho_t)h_t, \tag{16}$$

where

$$\rho_t = \frac{\mu}{4L_t}\left(\frac{\text{PWidth}(\mathcal{A})}{\operatorname{diam}(\mathcal{A})}\right)^2 \textit{ for AdaAFW and AdaPFW,}$$

$$\rho_t = \frac{\mu}{L_t}\left(\frac{\text{mDW}(\mathcal{A})}{\operatorname{radius}(\mathcal{A})}\right)^2 \textit{ for AdaMP.}$$

The previous theorem gives a geometric decrease on good steps. Combining this theorem with the bound for the number of bad steps in (7), and noting that the sufficient decrease guarantees that the objective is monotonically decreasing, we obtain a global linear convergence for AdaAFW, AdaPFW and AdaMP.

3.5 Discussion

Non-convex objectives. Lacoste-Julien (2016) studied the convergence of FW assuming the linear subproblems are solved exactly ($\delta = 1$) and obtained a rate of the form (11) with $C_0 = \max\{2h_0, L\operatorname{diam}(\text{conv}(\mathcal{A}))^2\}$ instead. Both rates are similar, although our analysis is more general as it allows to consider the case in which linear subproblems are solved approximately ($\delta < 1$) and also gives rates for the Away-steps and Pairwise variants, for which no rates were previously known.

Theorem 2 also gives the first known convergence rates for a variant of MP on general non-convex functions. Contrary to the case of FW, this bound depends on the mean instead of the maximum of the Lipschitz estimate.

Convex objectives. Compared with (Jaggi, 2013), the primal-dual rates of Theorem 3 are stronger as they hold for the last iterate and not only for the minimum

over previous iterates. To the best of our knowledge, primal-dual convergence rates on the last iterate have only been derived in (Nesterov, 2017) and were not extended to approximate linear subproblems nor the Away-steps and Pairwise variants.

Compared to Nesterov (2017) on the special case of exact subproblems ($\delta = 1$), the rates of Theorem 3 are similar but with $\overline{L}_t$ replaced by L. Hence, in the regime $\overline{L}_t \ll L$ (as is often verified in practice), our bounds have a much smaller leading constant.

For MP, Locatello et al. (2018) obtain a similar convergence rate of the form $\mathcal{O}(1/(\delta^2 t))$, but with constants that depend on global properties of ∇f, instead of the adaptive, averaged Lipschitz estimate in our case.

Strongly convex objectives. For the FW variants, the rates are identical to the ones in (Lacoste-Julien and Jaggi, 2015, Theorem 1), with the significant difference of replacing L with the adaptive L_t in the linear rate factor, giving a larger per-iteration decrease whenever $L_t < L$. Our rates are the first also covering approximate subproblems for Away-Steps and Pairwise FW algorithms. It's also worth noticing that both Away-steps FW and Pairwise FW have only been previously analyzed in the presence of exact line-search (Lacoste-Julien and Jaggi, 2015). Additionally, unlike (Lacoste-Julien and Jaggi, 2015), we do not require a smoothness assumption on f outside of the domain. Finally, for the case of MP, we again obtain the same convergence rates as in (Locatello et al., 2017, Theorem 7), but with L replaced by L_t.

4 Benchmarks

We compared the proposed methods across three problems and three datasets. The three datasets are summarized in the table of Figure 1, where density denotes the fraction of nonzero coefficients in data matrix and where the last two columns are quantities that arise during the optimization of AdaPFW and shed light into their empirical value. In both cases t is the number of iterates until 10^{-10} suboptimality is achieved.

4.1 ℓ_1-constrained logistic regression

The first problem that we consider is a logistic regression with an ℓ_1 norm constraint on the coefficients of the form:

$$\underset{\|\boldsymbol{x}\|_1 \leq \beta}{\arg\min} \ \frac{1}{n} \sum_{i=1}^{n} \varphi(\boldsymbol{a}_i^\top \boldsymbol{x}, \boldsymbol{b}_i) + \frac{\lambda}{2} \|\boldsymbol{x}\|_2^2 \ , \tag{17}$$

where φ is the logistic loss. β is chosen to give approximately 1%, 20% of nonzero coefficients respectively. The linear subproblems in this case can be computed exactly ($\delta = 1$) and consist of finding the largest entry of the gradient. The ℓ_2 regularization parameter λ is always set to $\lambda = \frac{1}{n}$.

We applied this problem on two different datasets: Madelon and RCV1. We show the results in Figure 1, subplots A, B, C, D. In this figure we also show the performance of FW, Away-steps FW (AFW) and Pairwise FW (PFW), all of them using the step-size $\gamma_t = \min\left\{g_t L^{-1} \|\boldsymbol{d}_t\|^{-2}, \gamma_t^{\max}\right\}$, as well as the backtracking variants of Dunn (1980) and (Beck et al., 2015), which we denote D-FW and B-FW respectively.

4.2 Nuclear-norm constrained Huber regression

The second problem that we consider is collaborative filtering. We used the MovieLens 1M dataset, which contains 1 million movie ratings, and consider the problem of minimizing a Huber loss, as in (Mehta et al., 2007), between the true known ratings and a matrix $\boldsymbol{X}$. We also constrain the matrix by its nuclear norm $\|\boldsymbol{X}\|_* \leq \beta$, where β is chosen to give approximately 1% and 20% of non-zero singular values respectively. The problem is of the form:

$$\underset{\|\boldsymbol{X}\|_* \leq \beta}{\arg\min} \frac{1}{n} \sum_{(i,j) \in \mathcal{I}}^{n} L_\xi(\boldsymbol{A}_{i,j} - \boldsymbol{X}_{i,j}) \ , \tag{18}$$

where H_1 is the Huber loss, defined as

$$L_\xi(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \leq \xi, \\ \xi(|a| - \frac{1}{2}\xi), & \text{otherwise}. \end{cases}$$

The Huber loss is a quadratic for $|a| \leq \xi$ and grows linearly for $|a| > \xi$. The parameter ξ controls this tradeoff and was set to 1 during the experiments.

In this case, the AFW and PFW variants were not considered as they are not directly applicable to this problem as the size of the active set is potentially unbounded. The results of this comparison can be see in subplots E and F of Figure 1. We emphasize that the goal of this experiment is to compare different FW variants and not find the best method for matrix completion. For alternative approaches not based on FW see for instance (Mareček et al., 2017).

We comment on some observed trends from these results:

- **Importance of backtracking.** Across the different datasets, problems and regularization regimes we found that backtracking methods always perform better than their non-backtracking variant.
- **Pairwise FW.** AdaPFW shows a surprisingly good performance when it is applicable, specially in the

Dataset	#samples	#features	density	$\overline{L}_t/L$	$(t - N_t)/t$
Madelon (Guyon et al., 2008)	4400	500	1.	3.3×10^{-3}	5.0×10^{-5}
RCV1 (Lewis et al., 2004)	697641	47236	10^{-3}	1.3×10^{-2}	7.5×10^{-5}
MovieLens 1M (Harper and Konstan, 2015)	6041	3707	0.04	1.1×10^{-2}	–

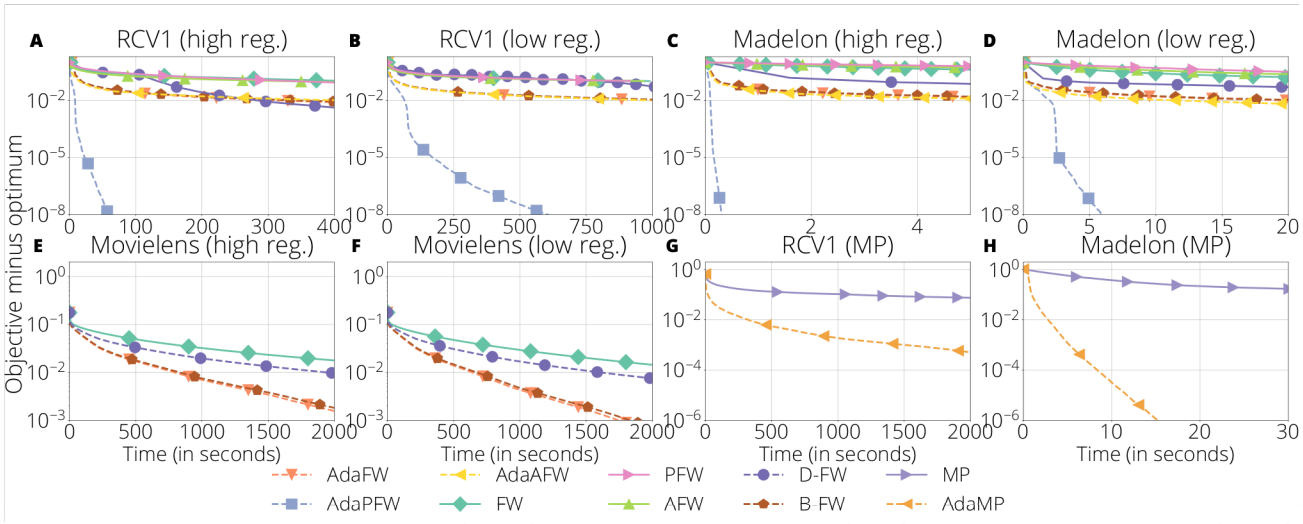


Figure 1: **Top table**: description of the datasets. **Bottom figure**: Benchmark of different FW and MP variants. The variants with backtracking line-search proposed in this paper are in dashed lines. Problem in A, B, C, D = logistic regression with ℓ_1-constrained coefficients, in E, F = Huber regression with on the nuclear norm constrained coefficients and in G, H = unconstrained logistic regression (MP variants). In all the considered datasets and regularization regimes, backtracking variants have a much faster convergence than others.

high regularization regime. A possible interpretation for this is that it is the only variant of FW in which the coefficients associated with previous atoms are not shrunk when adding a new atom, hence large step-sizes are potentially even more beneficial as coefficients that are already close to optimal do not get necessarily modified in subsequent updates.

- $\overline{L}_t$ **vs** L**.** We compared the average Lipschitz estimate $\overline{L}_t$ and the L, the the gradient's Lipschitz constant. We found that across all datasets the former was more than an order of magnitude smaller, highlighting the need to use a local estimate of the Lipschitz constant to use a large step-size.
- **Bad steps.** Despite the very pessimistic bounds obtained for the number of bad steps in the previous section, we observe that in practice these are extremely rare events, happening less than once every 10,000 iterations.

5 Conclusion and Future Work

In this work we have proposed and analyzed a novel adaptive step-size scheme that can be used in projection-free methods such as FW and MP. The method has minimal computational overhead and does not rely on any step-size hyperparameter (except for an initial estimate). Numerical experiments show large computational gains on a variety of problems.

A possible extension of this work is to develop backtracking step-size strategies for randomized variants of FW such as (Lacoste-Julien et al., 2013; Kerdreux et al., 2018; Mokhtari et al., 2018), in which there is stochasticity in the linear subproblems.

Another area of future research is to improve the convergence rate of the Pairwise FW method. Due to the very pessimistic bound on its number of bad steps, there is still a large gap between its excellent empirical performance and its known convergence rate. Furthermore, convergence of Pairwise and Away-steps for an infinite $\mathcal{A}$, such as the trace norm ball, is still an open problem.

Acknowledgements

The authors would like to thank Vlad Niculae and Courtney Paquette for valuable feedback on the manuscript.

References

Amir Beck, Edouard Pauwels, and Shoham Sabach. The cyclic block conditional gradient method for convex optimization problems. *SIAM Journal on Optimization*, 2015.

Dimitri P Bertsekas. *Nonlinear programming*. Athena Scientific, 1999.

Michael D Canon and Clifton D Cullum. A tight upper bound on the rate of convergence of Frank-Wolfe algorithm. *SIAM Journal on Control*, 1968.

Vladimir Demyanov and Aleksandr Rubinov. The minimization of a smooth convex functional on a convex set. *SIAM Journal on Control*, 1967.

Joseph C Dunn. Convergence rates for conditional gradient sequences generated by implicit step length rules. *SIAM Journal on Control and Optimization*, 1980.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics (NRL)*, 1956.

Dan Garber and Elad Hazan. A linearly convergent conditional gradient algorithm with applications to online and stochastic optimization. *arXiv preprint arXiv:1301.4666*, 2013.

Jacques Guélat and Patrice Marcotte. Some comments on Wolfe's away step. *Mathematical Programming*, 1986.

Isabelle Guyon, Steve Gunn, Masoud Nikravesh, and Lofti A Zadeh. *Feature extraction: foundations and applications*, volume 207. Springer, 2008.

F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2015.

Martin Jaggi. Revisiting Frank-Wolfe: projection-free sparse convex optimization. In *International Conference on Machine Learning*, 2013.

Thomas Kerdreux, Fabian Pedregosa, and Alexandre d'Aspremont. Frank-Wolfe with Subsampling Oracle. In *Proceedings of the 35th International Conference on Machine Learning*. PMLR, 2018.

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

Simon Lacoste-Julien and Martin Jaggi. On the global linear convergence of Frank-Wolfe optimization variants. In *Advances in Neural Information Processing Systems*, 2015.

Simon Lacoste-Julien, Martin Jaggi, Mark Schmidt, and Patrick Pletscher. Block-Coordinate Frank-Wolfe Optimization for Structural SVMs. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

Evgeny S Levitin and Boris T Polyak. Constrained minimization methods. *USSR Computational mathematics and mathematical physics*, 1966.

David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. RCV1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 2004.

Francesco Locatello, Rajiv Khanna, Michael Tschannen, and Martin Jaggi. A Unified Optimization View on Generalized Matching Pursuit and Frank-Wolfe. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017.

Francesco Locatello, Anant Raj, Sai Praneeth Karimireddy, Gunnar Raetsch, Bernhard Schölkopf, Sebastian Stich, and Martin Jaggi. On Matching Pursuit and Coordinate Descent. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 1993.

Jakub Mareček, Peter Richtárik, and Martin Takáč. Matrix completion under interval uncertainty. *European Journal of Operational Research*, 2017.

Bhaskar Mehta, Thomas Hofmann, and Wolfgang Nejdl. Robust collaborative filtering. In *Proceedings of the 2007 ACM conference on Recommender systems*, 2007.

BF Mitchell, Vladimir Demyanov, and VN Malozemov. Finding the point of a polyhedron closest to the origin. *SIAM Journal on Control*, 1974.

Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Stochastic Conditional Gradient Methods: From Convex Minimization to Submodular Maximization. *arXiv*, April 2018.

Yurii Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 2013.

Yurii Nesterov. Complexity bounds for primal-dual methods minimizing the model of objective function. *Mathematical Programming*, 2017.

John Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998.

Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alex Smola. Stochastic Frank-Wolfe methods for nonconvex optimization. In *54th Annual Allerton Conference on Communication, Control, and Computing*, 2016.