Accelerating conditional gradient methods

Thomas Kerdreux

PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'École Normale Supérieure

Accelerating Conditional Gradient Methods

Soutenue par
Thomas Kerdreux
Le 30 Juin 2020

École doctorale n°386

École Doctorale De Sciences Mathématiques de Paris

Spécialité

Mathématiques Appliquées

Composition du jury :

Francis Bach Researcher, INRIA Paris	*Président*
Zaid Harchaoui Associate Professor, University of Washington	*Rapporteur*
Mikael Johansson Professor, KTH	*Rapporteur*
Martin Jaggi Assistant Professor, EPFL	*Examinateur*
Sebastian Pokutta Professor, Technische Universität Berlin	*Examinateur*
Alexandre d'Aspremont Researcher, CNRS	*Directeur de thèse*

1794
ENS | PSL

"Chacune de ces machines, d'une manière ou d'une autre, ajoute à la puissance matérielle de l'homme, c'est-à-dire à sa capacité dans le bien comme dans le mal."

Georges Bernanos, *La France contre les robots.*

À ma mère et à mon père pour leur amour.

Résumé

Les méthodes de Gradient Conditionnel, ou algorithmes de Frank-Wolfe, sont des méthodes itératives du premier ordre utiles pour résoudre des problèmes d'optimisation sous contraintes. Elles sont utilisées dans de nombreux domaines comme l'apprentissage statistique, le traitement du signal, l'apprentissage profond, la géométrie algorithmique et bien d'autres encore. Ces algorithmes décomposent la minimisation d'une fonction non-linéaire en une série de sous problèmes plus simples. Chacun de ces sous-problèmes revient à minimiser une fonction linéaire sous les contraintes, l'oracle de minimisation linéaire. De nombreuses variantes de ces algorithmes existent qui cherchent à s'adapter au mieux aux structures particulières des problèmes d'optimisation sous-jacents. Ainsi de nombreuses directions de recherches restent ouvertes quant à l'analyse et la conception de nouveaux algorithmes de ce type, notamment pour l'apprentissage automatique.

Notre première contribution est de proposer et d'analyser de nouveaux schémas algorithmiques qui s'adaptent à un certain type d'hypothèses génériques. Ces dernières quantifient le comportement de la fonction près des solutions du problème d'optimisation. L'analyse de ces schémas d'algorithmes révèle des taux de convergence qui s'interpolent entre les taux classiques sous-linéaires en $\mathcal{O}(1/T)$ et les taux de convergence linéaire. Ces résultats montrent aussi que les algorithmes de Frank-Wolfe s'adaptent *facilement* à ce genre d'hypothèse puisque l'algorithme n'a pas besoin de connaître les paramètres qui contrôlent les hypothèses structurelles supplémentaires pour *accélérer*.

Notre seconde contribution s'inscrit dans une question de recherche encore ouverte. Les algorithmes de Frank-Wolfe peuvent accélérer le taux de convergence $\mathcal{O}(1/T)$ quand l'ensemble de contraintes est un polytope ou un ensemble fortement convexe. Pour quel autre type de contraintes existe-t-il une version de Frank-Wolfe avec des taux accélérés? Ici nous montrons que l'uniforme convexité, qui généralise la forte convexité, permet d'accélérer l'algorithme de Frank-Wolfe, là encore de manière adaptative. Plus généralement, cela signifie que c'est la courbure des ensembles de contraintes – et pas seulement une quantification spécifique telle que la forte convexité – qui peut accélérer les algorithmes de Frank-Wolfe.

Pour notre troisième contribution, nous proposons des versions des algorithmes de Frank-Wolfe où l'oracle de minimisation linéaire est résolu sur des sous-ensembles aléatoires de l'ensemble de contraintes initial tout en conservant, en espérance, les même taux de convergence asymptotiques. Bien que ces algorithmes ne conservent pas toutes les propriétés classiques des algorithmes de Frank-Wolfe, ce résultat étend les résultats de descente par blocs de coordonnées qui s'appliquent lorsque l'ensemble de contraintes est le produit cartésien d'ensembles plus simples.

Finalement notre quatrième contribution vise à raffiner théoriquement les taux dans le lemme de Carathéodory approximé de sorte à prendre en compte une mesure de la variance,

dans une norme de Banach, des atomes formant l'enveloppe convexe en question. Ce résultat repose sur un extension des inégalités de concentration de type Serfling, c'est-à-dire de tirage avec remplacement. Nous appliquons ce résultat pour des versions approximées du théorème de Shapley-Folkmann.

En appendice nous relatons des recherches faites en parallèle du sujet principal de recherche.

Abstract

Conditional gradient algorithms, a.k.a. the Frank-Wolfe algorithms, are first-order iterative methods designed for solving large-scale constrained optimization problems. These are used in a variety of modern applied fields such as machine learning, signal processing, deep learning, computational geometry and many others. They break down non-linear constrained problems in a series of small subproblems. Each of these requires at worst to minimize a linear function over the constraint set, the so-called Linear Minimization Oracles (LMO). This framework encompasses a growing series of algorithms that seek to adapt to particular structures of the optimization problem. Many open questions remain in the convergence analysis and design of such algorithms.

Our first contribution is to derive and analyze new Frank-Wolfe algorithms assuming specifically designed Hölderian Error Bounds. Our algorithms exhibit accelerated convergence rates without knowledge of the error bound parameters, i.e. they are *adaptive*. Our analysis also provides the first interpolated convergence rates between standard sublinear rates $\mathcal{O}(1/T)$ and linear convergence rates.

Our second contribution is focused on finding families of constraint sets for which Frank-Wolfe algorithms have accelerated convergence rates, outside of the classical scenarios where the set is either a polytope or a strongly convex set. We prove that the original Frank-Wolfe algorithm enjoys adaptive accelerated rates when the set is uniformly convex. This structural assumption subsumes strong-convexity and quantifies the curvature regimes of classical constraint sets that strong convexity does not capture.

In our third contribution, we design Frank-Wolfe algorithms where the Linear Minimization Oracle is only solved on random subsets of the constraints while retaining, in expectation, classical convergence rates. Although it does not maintain all benefits of Frank-Wolfe algorithms, the method extends block-coordinate type algorithms, which only converge when the constraint set is the cartesian products of simpler sets.

Our fourth contribution focuses on refining the bounds of the approximate Carathéodory lemma by taking into account the variance of the convex hull as measured with general Banach norms. This result relies on an extension of a Serfling concentration inequality type to Banach spaces. We use this new approximate Carathéodory lemmas to refine approximate versions of the Shapley-Folkman theorem.

In the appendix, we relate some orthogonal research directions that were developed in parallel with the principal research subject. The first one stems from a work carried out with Antoine Recanati where we extend a result of Atkins to a multi-dimensional setting. The second is also a collective contribution with Louis Thiry and Vivien Cabannes where we use the pretence of making interactive paintings between artists and machines powered with what-people-call-artificial-intelligence algorithms to offer a demystifying perspective on these.

Remerciements

Je remercie en premier lieu mon directeur de thèse Alexandre d'Aspremont. Il a été un merveilleux guide. Sa clarté intellectuelle et son enthousiasme permanent ont été de formidables stimulants durant ces trois années de thèse. Je le remercie aussi très particulièrement pour l'équilibre qu'il m'a donné entre liberté et exigence dans la recherche. Ces années ont pu donc être riches d'apprentissages scientifiques mais m'ont aussi donné le temps de mieux comprendre le monde dans lequel nous vivons.

Je voudrais aussi exprimer ma plus sincère gratitude aux membres du jury: Francis Bach, Martin Jaggi, Mikael Johansson et Zaid Harchaoui. Plus spécialement encore, je remercie Sébastian Pokutta de m'avoir accueilli dans son groupe à Berlin pendant un mois. Cela a été un moment très instructif et agréable.

Je remercie aussi chaleureusement Francis Bach pour l'environnement serein, multi-disciplinaire et de grande qualité qu'il a créé au quatrième étage de l'INRIA de Gare de Lyon. Je n'aurais pas pu espérer un meilleur groupe pour faire ma thèse et j'ai beaucoup appris de la diversité des opinions et des personnalités qui se sont croisées au quatrième étage. J'ai aussi beaucoup apprécié la proximité immédiate avec l'équipe Willow. J'ai pu ainsi poser toutes mes questions à des spécialistes de vision par ordinateur (merci Yana !), robotique ou encore d'apprentissage par renforcement (merci Yann !), ces disciplines étant amenées à jouer (ou jouant déjà) un rôle majeur dans les révolutions technologiques sur lesquelles nous n'aurons pas totalement notre mot à dire.

Je voudrais aussi remercier la Fondation Biermans-Lapôtre et ces membres, Catherine Ruberti, Diane Miller, Claude Gonfroid et bien d'autres encore. Ca a été un environnement exceptionnel, tant pour son cadre que ces habitants, pour mes deux années de thèse.

Mon arrivée dans le monde de la recherche s'est faite grâce à Éric Matzner-Lober lors de ma troisième année à l'École Polytechnique. Il m'a ouvert à ce monde avec beaucoup de pédagogie, sympathie et énergie. Depuis lors, il continue toujours à me donner de précieux conseils dans tous les domaines de la vie. Je dois aussi le remercier, et c'est presque anecdotique par rapport à tout ce qu'il m'a apporté, de m'avoir prêté son appartement parisien pendant la crise du Covid-19. Ce fut un environnement très calme, absolument solitaire mais justement idéal pour écrire cette thèse.

Éric m'a aussi présenté à Nicolas Hengartner qui m'a accueilli plusieurs étés dans son groupe de biologie théorique au Los Alamos National Laboratory aux États-Unis. L'expérience croisée de différentes cultures scientifiques a été très instructive. En plus d'être un ami, Nick m'a fait découvrir de nombreux sujets de recherche appliquée et théorique avec beaucoup d'intuition, d'énergie et de rigueur. Je le remercie aussi des rencontres qu'il m'a permis de faire comme Sylvain Sardy, Marian Anghel et bien d'autres encore! J'espère que l'année prochaine à Los Alamos nous pourrons enfin finir au moins une partie de ce que nous avons commencé ensemble,

et ramasser toujours plus de lobster mushrooms !

Je suis aussi profondément reconnaissant à CFM et à Jean-Philippe Bouchaud d'avoir créé et organisé sa fondation pour la recherche qui m'a octroyé une bourse de thèse. Cela m'a donné une très grande liberté. Je me suis donc efforcé chaque jour d'en tirer le meilleur parti. J'ai aussi beaucoup aimé ces journées pour la recherche organisées par CFM où Jean-Philippe Bouchaud assistait personnellement aux présentations de tous "ses" doctorants, mille mercis !

Durant ces trois années de thèse, j'ai effectué un monitorat d'enseignement pour le master de science des données X-HEC et je voudrais donc remercier Julie Josse de m'avoir fait pleinement confiance dans cette tâche. Cela a été une expérience très instructive pédagogiquement et scientifiquement car c'est une chose de comprendre les concepts pour faire de la recherche mais cela en est aussi une autre de les expliquer simplement. Merci aussi à Stéphane Canu pour toutes nos discussions et de m'avoir fait confiance pour donner le cours d'apprentissage par renforcement.

Je voudrais aussi remercier Fabian Pedregosa avec qui j'ai écrit mon premier papier sur les algorithmes de Frank-Wolfe. Ca a été un plaisir de travailler avec lui ; la distance a fait malheureusement que l'expérience ne s'est pas encore répétée, mais cela ne saurait tarder. Merci aussi à Antoine Recanati pour notre collaboration et ses très beaux écrits. Je voudrais aussi remercier mes co-bureaux, Vincent Roulet, Antoine Recanati, Damien Scieur, Mathieu Barré, Gregoire Mialon et Radu Dragomir de m'avoir supporté pendant le temps que nous avons passé ensemble confinés – c'est un mot à la mode en ce moment – dans notre vaste bureau. Merci aussi à tous les autres membres de Willow-Sierra et occupants du quatrième étage, Thomas Eboli, Pierre-Louis Guhur, Yana Hasson, Yann Labbé, Julia Peyre, Adrien Taylor et tous les autres !! Je voudrais aussi remercier Charles Valeyre avec qui j'ai eu de nombreuses discussions et avec qui nous avons pu avoir un avant goût à ce que signifiait de monter une entreprise. Merci aussi à Chiara Musolini, Thomas Lartigue, Daniel Reiche, Laura Spector et d'autres encore dont les qualités humaines et intellectuelles ont enrichi ces dernières années.

Enfin je suis heureux de pouvoir travailler avec mon ami de longue date, Louis Thiry, sur ces projets où, en créant des processus d'interaction entre machines et artistes, nous espérons témoigner de notre époque charnière. Nous vivons dans un nouveau monde, où les machines et les algorithmes n'ont jamais été aussi proches de l'Homme. Ils risquent subrepticement de nous redéfinir individuellement et collectivement. Merci aussi à Vivien Cabannes dans ce projet mais aussi pour son amitié précieuse (et tumultueuse) dont j'apprécie toujours la sincérité et l'exigence de clarté. Merci aussi à Léa Saint-Raymond pour son enthousiasme, son énergie et son vaste savoir multi-disciplinaire qui nous éclaire dans ce projet. Merci aussi à Tina Campana et Charly Ferrandes pour leur talent et leur confiance passés et à venir. Merci aussi à mon père d'avoir prêté ses talents de peintre dans ces jeux interactifs.

Jean-François Gribinski a aussi profondément influencé et accéléré par ses écrits et ses mots ma vision du monde. Il a donc contribué très largement à ma maturation intellectuelle.

Je suis aussi heureux de bientôt collaborer avec Bertrand Rouet-Leduc et Claudia Hulbert au Los Alamos National Laboratory. J'ai hâte de travailler avec vous deux, de découvrir un nouveau domaine scientifique, des météorites et peut-être aussi toutes les sources chaudes de la Caldera !

Enfin je voudrais remercier tout spécialement Lise-Marie Bivard qui a été un extraordinaire soutien et interlocuteur pour toutes les démarches administratives durant ces trois années. Merci aussi à Eric Sinaman pour son efficacité dans la gestion de ma soutenance de thèse.

Contributions and thesis outline

This dissertation primarily focuses on designing or analyzing new conditional gradient algorithmic schemes, a.k.a. Frank-Wolfe algorithms. We consider the general constrained optimization problem

$$\underset{x \in \mathcal{C}}{\text{minimize}}\ f(x), \tag{1}$$

where $\mathcal{C}$ is a compact convex set and f is a differentiable convex function. A crucial feature our dissertation demonstrates is that Frank-Wolfe algorithms are *adaptive* to several types of structural assumptions on the optimization problem.

Chapter 1: This chapter offers a review of Frank-Wolfe algorithms. We survey Frank-Wolfe algorithm and point to existing convergence results and applications. In this chapter, we do not provide new results.

Chapter 2: This chapter focuses on designing and analyzing versions of Frank-Wolfe *adaptive* to error-bound type conditions. We notably tailor *error bounds* assumptions for the Frank-Wolfe algorithms. We then show that *restarted* versions of Frank-Wolfe enjoy new sublinear convergence rates without specific knowledge of the error bounds parameters. In other words, there exist Frank-Wolfe algorithms *adaptive* to generic structural assumptions on the geometry of the problem around its optimal solutions.

Chapter 3: In this chapter, we focus on the original Frank-Wolfe algorithm. We show that, under appropriate assumptions on f, it enjoys accelerated convergence when the constraint set is uniformly convex. This is a generic quantification of the *curvature* of a set subsuming strong-convexity. For instance, the ℓ_p balls are uniformly convex for all $p > 1$, but strongly convex for $p \in]1, 2]$ only. Hence, our analysis non-trivially generalizes the various rates under strong-convexity assumptions. It is the curvature of the constraint sets – not just their strong convexity – that leads to accelerated convergence rates for Frank-Wolfe. These conclusions also highlight that the Frank-Wolfe algorithm is *adaptive* to much more generic constraint set structures, thus explaining faster empirical convergence. Finally, we also show accelerated convergence rates when the set is only locally uniformly convex and provide similar results in online linear optimization.

Chapter 4: Here we propose *randomized* – or *subsampled* – variants of the Frank-Wolfe algorithms, which solve linear minimization problems over a small *subset* of the original domain. We show that, in expectation, the randomization does not affect the various asymptotic convergence rates. We obtain a $\mathcal{O}(1/t)$ sublinear convergence rate for *randomized* Frank-Wolfe

and a linear convergence rates for *randomized* away-step Frank-Wolfe. While subsampling reduces the convergence rate by a constant factor, the cost of the linear minimization step can be a fraction of the deterministic versions, especially when the data is streamed. We illustrate computational gains on regression problems, involving both ℓ_1 and latent group lasso penalties.

Chapter 5: This last chapter steps slightly aside from the analysis or designing of Frank-Wolfe algorithms. Here, we sought to improve results for the approximate Shapley-Folkmann [d'Aspremont and Colin, 2017]. The results crucially depend on the application of the approximate Carathéodory lemma in a regime where the number of atoms to approximate any point of a convex hull is very close to $d+1$, where d is the ambient dimension. Because classical proof of the approximate Carathéodory lemma relies on sampling results, we call it the *high-sampling* regime. We hence devise concentration inequalities for that regime, a.k.a. Serfling-type concentration inequalities, and for general Banach spaces in order to handle non-Hilbertian norms. This notably allows stating a version of the approximate Carathéodory lemma using a notion of variance of the atoms, not just the diameter of the convex hull.

Appendix A: Here we extend a classical result of Atkins et al. [1998] for seriation problems. These consist in constructing an ordering for a set of objects given the similarity matrix, *i.e.* the matrix of their pair-wise measures of similarity. Atkins et al. [1998] recover an ordering from that of the second eigenvector of the Laplacian of the similarity matrix. In other words, this consists in retrieving the appearing order from the 1-dimensional embedding stemming from the Laplacian. Some works eventually considered a similar strategy for the 2-dimensional Laplacian embedding. Here we show that more generally, increasing the size of the embedding may maintain a filamentary structure. We provide algorithms to recover the ordering from these filamentary structures, and we experimentally show that our method is much more resilient to the noise stemming from the inexact pair-wise measures of similarity. For some class of matrix typically found in seriation, we also give theoretical guarantees.

Publications related to this manuscript are listed below.

- **Chapter** 2 is based on the article [Kerdreux et al., 2019]: Thomas Kerdreux, Alexandre d'Aspremont, and Sebastian Pokutta. "Restarting Frank-Wolfe." *The 22nd International Conference on Artificial Intelligence and Statistics.* 2019.

- **Chapter** 3 is based on the article [Kerdreux et al., 2020a]: Thomas Kerdreux, Alexandre d'Aspremont, and Sebastian Pokutta. "Projection-Free on Uniformly Convex Sets." 2020. *arXiv preprint arXiv:2004.11053.*

  This chapter is a generalization and an extension of a partial result given in the last section of [Kerdreux et al., 2018a]: Thomas Kerdreux, Alexandre d'Aspremont, and Sebastian Pokutta. "Restarting Frank-Wolfe: Faster Rates Under Hölderian Error Bounds." *arXiv preprint arXiv:1810.02429v3* 2019.

- **Chapter** 4 is based on the article [Kerdreux et al., 2018b]: Thomas Kerdreux, Fabian Pedregosa, and Alexandre d'Aspremont. "Frank-Wolfe with Subsampling Oracle." *International Conference on Machine Learning.* 2018.

- **Chapter** 5 is partially based on the article [Kerdreux et al., 2017]: Thomas Kerdreux, Igor Colin, and Alexandre d'Aspremont. "An Approximate Shapley-Folkman Theorem." *arXiv preprint arXiv:1712.08559v3* 2017.

- **Appendix** A is the article [Recanati et al., 2018a]: Antoine Recanati, Thomas Kerdreux, Alexandre d'Aspremont. "Reconstructing Latent Orderings by Spectral Clustering." 2018. *arXiv preprint arXiv:1807.07122*.

- **Appendix** B is a link to the following two papers [Cabannes et al., 2019, Kerdreux et al., 2020b]:

  Vivien Cabannes, Thomas Kerdreux, Louis Thiry, Tina Campana, Charly Ferrandes. "Dialog on a canvas with a machine." 2019. Presented at the workshop of *Creativity and design* at *NeurIPS 2019*.

  Thomas Kerdreux, Louis Thiry, Erwan Kerdreux. "Interactive Neural Style Transfer with Artists." *arXiv preprint arXiv:2003.06659* and accepted as an oral presentation at *The 11nd International Conference on Computational Creativity.* 2020.

Contents

Chapter 1

Conditional Gradient Algorithms

A constrained optimization problem seeks to find the extremal values of a function $f : \mathbb{R}^d \to \mathbb{R}$ when x belongs to a set $\mathcal{C} \subset \mathbb{R}^d$. Such optimization problems are pervasive in domains like machine learning, signal processing, economic, computational geometry, deep learning and many others. The problems we consider all along this dissertation are of the form

$$\min_{x \in \mathcal{C}} f(x), \tag{1.1}$$

where f is a convex differentiable function and the constraint set $\mathcal{C}$ is a compact convex set. Since there is usually no analytical formula to describe the solutions of (1.1), numerous algorithmic methods have been designed to find approximate solutions by iteratively refining a sequence of points (x_t). With convexity assumptions, the difficulty is not to build algorithms that converge, but to design ones that *best* reduce the number of iterations and their computational cost to obtain an approximate solution to (1.1).

At each iteration, an algorithm exploits some knowledge of f or $\mathcal{C}$ that is accessed via *oracles*. Most of the algorithms use *homogeneous* type of oracles at each iteration. As such, one can typically classify them according to the structural knowledge required by these oracles. For instance, *first-order* algorithms involve computations of the gradient of f. There are also zero-order, second-order, or first-order stochastic algorithms that respectively require the computation of the function values, the Hessians, or stochastic approximation of the gradients.

In constrained optimization problems, algorithms also need to access a set-related oracle. For large-scale instance of (1.1), there are typically two commonly used paradigms: proximal operators or linear minimization oracles (LMO) on $\mathcal{C}$. A proximal operator more or less requires to minimize a quadratic function over the constraint set $\mathcal{C}$. Each type of oracle has its realm of efficiency. In this dissertation, we focus on first-order algorithms relying on Linear Minimization Oracles, known as the Frank-Wolfe algorithms or conditional gradient algorithms.

To avoid confusion, we identify an algorithm to this family when an iteration requires at worst to solve a linear minimization problem over the original domain, a subset of the domain or a *reasonable* modification of the domain, *i.e.* a change that does not stealthily amount to a proximal operation.

Analyzing algorithms properties under various structures of the optimization problems helps to build a practitioner synopsis of the many existing algorithms. Conversely, it is crucial to design new algorithms that best use the information given by the oracles in order to reduce the number of inefficient operations.

A key concept we demonstrate in this dissertation is that the Frank-Wolfe algorithms are adaptive to various types of structural assumptions. Our contributions show that there exist Frank-Wolfe variants that exhibit accelerated convergence rates under various parametric structural assumptions, without requiring knowledge of these parameters.

1.1 Conditional Gradient Framework

Here we present a partial review of the Frank-Wolfe algorithms. We will interchangeably call these algorithms *Frank-Wolfe* or *conditional gradient* algorithms. Our introduction notably focuses on explaining the known *adaptive* properties of Frank-Wolfe algorithms.

In this section, we first survey some of the critical features of the original Frank-Wolfe algorithm and important related notions in convex analysis and geometry. In Section 1.2, we collect classical convergence results of this algorithm, which we will be referring to during this dissertation. In Section 1.3, we summarize various *corrective* versions of the Frank-Wolfe algorithm and the rich recent literature analyzing various aspects of these methods. Finally, in Section 1.4, we survey the wealth of Frank-Wolfe algorithms exploring different structural settings besides the smooth convex minimization over a compact convex set. We also point to the many domains leveraging Frank-Wolfe algorithms.

Notations. $\mathcal{C}$ will always stand for a convex set and we set aside d for the ambient dimension of $\mathcal{C}$ in finite normed spaces. When working in an Hilbertian space we write $\langle , \rangle$ its scalar product. The polar of a convex set $\mathcal{C}$ is defined as $\mathcal{C}^\circ \triangleq \left\{ y : \langle x, y \rangle \leq 1, \ \forall x \in \mathcal{C} \right\}$. For a matrix $M \in \mathbb{R}^{n \times m}$, its p-Schatten norm is defined as the ℓ_p norm of the vector (σ_i) of its singular values, $||M||_{S(p)} \triangleq \left(\sum_{i=1}^{\max\{n,m\}} \sigma_i^p \right)^{1/p}$. For a set $\mathcal{C}$, we note $\mathrm{Aff}(\mathcal{C})$ the affine hull of $\mathcal{C}$, $\mathrm{Conv}(\mathcal{C})$ its convex hull and $\mathrm{Co}(\mathcal{C})$ its conic hull.

1.1.1 The Frank-Wolfe Algorithm

There are now many different variants of Frank-Wolfe algorithms or conditional gradient methods [Levitin and Polyak, 1966, §6]. Here, we review the original Frank-Wolfe algorithm [Frank and Wolfe, 1956] as it captures many of the properties and concepts that will be used all along this dissertation. It is a first-order iterative method. At each iteration, it finds an element in the domain $\mathcal{C}$ that minimizes the linear approximation of the objective function. It then performs a convex update between this element and the current iterate.

Each iteration of the Frank-Wolfe algorithm hence relies on the minimization of a linear function over the domain $\mathcal{C}$, called the Linear Minimization Oracle and defined as follows for $h \in \mathbb{R}^d$.

$$\mathrm{LMO}_{\mathcal{C}}(h) \in \operatorname*{argmin}_{v \in \mathcal{C}} \langle h, v \rangle. \tag{1.2}$$

The Frank-Wolfe methods are called *projection-free* as opposed to projected gradient descent or proximal methods. These other types of algorithms update their iterates along feasible directions that do not necessarily maintain the iterates in the domain $\mathcal{C}$. As such, they require some *projection oracles*, which computational cost is that of minimizing a quadratic function over the domain $\mathcal{C}$. When the domain $\mathcal{C}$ is a polytope, the Frank-Wolfe iterations rely on Linear Programming (LP) subproblems. In contrast, proximal methods rely on Quadratic

Programming (QP). Another example is the case of the nuclear norm (the ℓ_1 norm of a matrix singular values). In that case, the Linear Minimization Oracle requires the knowledge of the leading singular value as opposed to a projection oracle, which relies on the computation of the full singular value decomposition (SVD).

We now state the Frank-Wolfe algorithm with the three main types of line-search in Algorithm 9.

Algorithm 1 The Frank-Wolfe Algorithm

Input: $x_0 \in \mathcal{C}$, $\epsilon > 0$.

1: **for** $t = 0, 1, \ldots, T$ **do**
2: $v_t \in \underset{v \in \mathcal{C}}{\operatorname{argmax}} \langle -\nabla f(x_t); v - x_t \rangle = \mathrm{LMO}_{\mathcal{C}}(\nabla f(x_t))$ ▷ Linear Minimization Oracle
3: **if** $\langle -\nabla f(x_t); v_t - x_t \rangle < \epsilon$ **then**
4: **return** x_t
5: **end if**
6: Variant 1: $\gamma_t = 1/(t+1)$ ▷ Determinist step-size
7: Variant 2: $\operatorname{argmin}_{\gamma \in [0,1]} \gamma \langle v_t - x_t; \nabla f(x_t) \rangle + \gamma^2 L ||v_t - x_t||^2/2$ ▷ Short step-size
8: Variant 3: $\operatorname{argmin}_{\gamma \in [0,1]} f(x_t + \gamma(v_t - x_t))$ ▷ Exact line-search
9: $x_{t+1} = (1 - \gamma_t) x_t + \gamma_t v_t$
10: **end for**

In (1.2), one can always choose an extreme point of $\mathcal{C}$ among the solutions of the linear minimization. This is particularly important as the iterates of the Frank-Wolfe algorithm are convex combinations of these outputs. As such, there is a close link between the extremal structure of $\mathcal{C}$ and structural properties of the iterates x_t. For instance, if the extreme points of $\mathcal{C}$ are low-rank matrices, then the early algorithm iterates (if properly initialized) will also be low-rank-matrices.

In line 4 of Algorithm 9, a stopping criterion is a by-product of the Linear Minimization Oracle. Indeed by optimality of v_t and convexity of f, we have

$$\langle -\nabla f(x_t); v_t - x_t \rangle \geq \langle -\nabla f(x_t); x^* - x_t \rangle \geq f(x_t) - f(x^*).$$

The quantity $\langle -\nabla f(x_t); v_t - x_t \rangle$ is often referred as the Frank-Wolfe gap [Jaggi, 2013] and denoted $g(x_t)$.

1.1.2 Useful Notions of Convex Geometry

Most of the concepts here are classic elements of convex analysis [Rockafellar, 1970b] that will be useful for this dissertation. Extreme points connect to the structural properties of the Frank-Wolfe iterates. The faces of the convex domain $\mathcal{C}$ and associated cones are related to the analysis and design of Frank-Wolfe algorithms. We also define two crucial families of convex domains.

Extreme Points of Convex Sets. A point x of a convex set $\mathcal{C}$ is an extreme point of $\mathcal{C}$ if and only if it cannot be written $\lambda y + (1 - \lambda) z$ with $x, y \in \mathcal{C}$ unless $x = y = z$ [Rockafellar, 1970b, §18]. We will refer to the *extremal structure* of $\mathcal{C}$ as the properties the extreme points of $\mathcal{C}$ might share. They are particularly important because there is always an extreme point of $\mathcal{C}$ in the solutions of the Linear Minimization Oracle.

This is a crucial aspect of conditional gradient algorithms as it allows to enforce specific structures on the iterates that approximate a solution of (1.1). The early iterates are sparse convex combinations of these extremes points, also called *atoms*. Each specific extremal structure of constraint sets $\mathcal{C}$ provides a different meaning to the sparsity of the iterates. For instance, with the ℓ_1 balls, the iterates of Frank-Wolfe are sparse in the classical sense, in terms of non-zero coordinates in the canonical basis. With the trace-norm balls, a.k.a. the nuclear balls, the matrix iterates are sparse convex combinations of low-rank matrices. Hence the first iterates are also low-rank matrices. The extremal structure of the convex set $\mathcal{C}$ directly controls the structure of the Frank-Wolfe iterates, a useful mechanism in many practical scenarios.

Carathéodory lemma states that for a point x in a convex subset of $\mathbb{R}^d$, there exists a representation of it as a convex combination of at most $d+1$ extreme points of $\mathcal{C}$. It means that, in theory, it is possible to maintain the Frank-Wolfe iterates as a convex combination of at most $d+1$ extreme points of $\mathcal{C}$.

For the specific case of the Frank-Wolfe algorithm, we can only ensure that after T iterations, the iterate will be a convex combination of at most T such extreme points. Most variants of Frank-Wolfe algorithms share or improve over this property, see *corrective* variants in Section 1.3.

Faces of Convex Sets. Extreme points are zero-dimensional faces of convex sets $\mathcal{C}$. A face F of $\mathcal{C}$ is a convex subset of $\mathcal{C}$ such that every line segment in $\mathcal{C}$ with a relative point in F has both endpoints in F [Rockafellar, 1970b, §18]. The dimension of the face is the dimension of its affine hull. While the extremal structure of $\mathcal{C}$, *i.e.* the zero-dimensional facial structure, controls the structure of the iterates, the general facial structure of $\mathcal{C}$ is more important in the analysis and design of Frank-Wolfe algorithms.

In many practical applications of Frank-Wolfe in machine learning, the constraint sets rarely exhibit pathological behaviours. As an arbitrary example, we never encountered a situation where the set of extreme points was not a closed set, which is however not true in general. Indeed, most of the examples have relatively simple structures such as polytopes, strongly-convex sets, uniformly convex sets, intersections of polytopes and strongly-convex sets or slightly more complex such as nuclear balls or some structured norm balls.

Polytope and Strongly Convex Sets. Polytopes and strongly convex sets are the two families of set for which there exist a Frank-Wolfe variants with both *enhanced* theoretical and empirical properties.

Polytopes are arguably the most common type of convex sets $\mathcal{C}$ appearing in practical applications. Polytopes are bounded sets that can either be represented as the intersection of several half-spaces, an *external* representation, or as the convex hull of a finite number of points (atoms) $\mathcal{A}$, an *internal* representation. Polytopes admit a finite number of extreme points and have a *homogeneous* facial structure, *i.e.* each face is also a polytope. In Section 1.3, we present the *corrective* or *away* versions of the Frank-Wolfe algorithm that were arguably designed for this type of structure.

Strongly convex sets is also an important family of structured sets. It is a specific quantification of the curvature of the boundary of some convex sets. In Chapter 3, we show that more general quantifications of curved sets can be leveraged in the context of the Frank-Wolfe algorithms.

Definition 1.1.1 (strongly-convex sets)**.** *A compact convex set $\mathcal{C}$ is strongly convex with respect to the norm $||\cdot||$ if and only if there exists $\alpha > 0$ such that for all $(x, y) \in \mathcal{C}$, all $\gamma \in [0, 1]$ and all $z \in B_{||\cdot||}(0, 1)$*

$$\gamma x + (1-\gamma)y + \alpha\gamma(1-\gamma)||x-y||^2 z \ \in \mathcal{C} \ . \tag{1.3}$$

In euclidean finite-dimension settings, we often say that $\mathcal{C}$ is a strongly convex set without specifying any norm. It is implicitly with respect to the euclidean norm.

Here, any point $x \in \partial\mathcal{C}$ is extreme, and the faces of a strongly convex set are the set itself and its extreme points. Others equivalent definitions of the strong convexity of a set exist but are not useful here [Weber and Reisig, 2013]. Note also that Definition 1.1.1 depends on a specific norm $||\cdot||$, impacting the value of the constant α. In finite dimension, because norms are equivalent, when a set is strongly convex for a given norm, it is with varying parameters α. This has an influence on some convergence rate of the Frank-Wolfe algorithm. In particular, it is not an affine invariant notion.

The strong-convexity of a set is especially interesting as it often brings a *quadratic structure* on the optimization problem (1.1) sufficient to accelerate the Frank-Wolfe algorithms, without additional *quadratic structure* on the function f (besides smoothness), see Section 1.2.

Convex Cones. A convex cone $\mathcal{K}$, is a set in $\mathbb{R}^d$ such that for any tuple $x, y \in \mathcal{K}$ and any tuple of non-negative coefficients (α, β), $\alpha x + \beta y \in \mathcal{K}$. Several type of natural convex cone appears in convex geometry [Rockafellar, 1970b, §2]. The situation is considerably simpler for practical analyses of Frank-Wolfe algorithms. The normal cone to a point $x \in \partial\mathcal{C}$ with respect to $\mathcal{C}$ is defined as

$$\mathcal{K}_{\mathcal{C}}(x) = \left\{h \text{ s.t. } \langle y - x; h\rangle \leq 0 \ \ \forall y \in \mathcal{C}\right\} \ . \tag{1.4}$$

At a point x of the boundary of $\mathcal{C}$, $\mathcal{K}_{\mathcal{C}}(x)$ gathers all the directions that are negatively correlated with all admissible direction to $\mathcal{C}$ at x. Normal Cone is a notion very closely related to Linear Minimization Oracles as a extreme point v^* solution of

$$\operatorname*{argmin}_{v \in \mathcal{C}} \langle h; v\rangle,$$

is such that $h \in \mathcal{K}_{\mathcal{C}}(v^*)$. Normal cones offer simple geometrical understanding of the behavior of a Frank-Wolfe algorithm, such as the *zig-zag* phenomenon [Wolfe, 1970, Guélat and Marcotte, 1986] or the improved convergence when the set is *curved*, see Figure **??** in Chapter 3. The analyses of Frank-Wolfe algorithms then often seek to summarize this geometrical perspective into a single algebraical formula. In particular, this is probably the reason why only arguably *homogeneous* type of structure of the constraint sets has been studied in the analyses of the Frank-Wolfe algorithms.

1.1.3 Useful Notions of Convex Analysis

We have just reviewed the important properties of the convex constraint sets that appear in optimization problem (1.1). Similarly, the convergences of the algorithms also depend on structural assumptions on the objective functions f in (1.1). Most of our work has been done in the context of convex differentiable functions which revolve around two main assumptions, L-smoothness and μ strong convexity. In Section 2.1 of Chapter 2, we review other types of assumptions which provide a way for convex function to interpolate and localize the convexity

and strong-convexity behaviors. In Chapter 2, we then define errors bounds for the specific setting of Frank-Wolfe algorithms and show how they can be leveraged.

Definition 1.1.2 (Convex function)**.** *A function is convex if for any distinct* $(x, y) \in \mathcal{C}$ *and* $\gamma \in [0, 1]$

$$f(\gamma x + (1-\gamma)y) \leq \gamma f(x) + (1-\gamma) f(y) \ . \tag{1.5}$$

When (1.5) holds strictly, the function is *stricly convex*, guaranteeing the unicity of a solution to (1.1). When the function f is differentiable, convexity implies that the function is lower bounded by its linear approximations. Strong convexity then strengthens this by requiring the function to be lower bounded by a quadratic approximation of it. For a differentiable function f we now state the strong convexity property in Definition 1.1.3.

Definition 1.1.3 (Strongly convex function)**.** *A differentiable function* f *on* $\mathcal{C}$ *is strongly convex (with respect to* $||\cdot||$*) if there exists* $\mu > 0$ *such that for any* $(x, y) \in \mathcal{C}$

$$f(y) \geq f(x) + \langle \nabla f(x); y - x \rangle + \frac{\mu}{2}||x - y||^2. \tag{1.6}$$

Conversely, the L-smoothness property means that the gradient is a L-Lipschitz function on $\mathcal{C}$. It then bounds the amplitude of the variations of the function and means that the function is upper bounded by a quadratic approximation of it, see Definition 1.1.4. For feasible direction methods, this upper bound directly controls the amount of primal decrease to hope for at each iteration.

Definition 1.1.4 (Smooth convex function)**.** *A differentiable function* f *on* $\mathcal{C}$ *is smooth (with respect to* $||\cdot||$*) if there exists* $L > 0$ *such that for any* $(x, y) \in \mathcal{C}$

$$f(y) \leq f(x) + \langle \nabla f(x); y - x \rangle + \frac{L}{2}||x - y||^2. \tag{1.7}$$

These properties can be localized. There is also a plethora of conditions and inequality one can derive from convex functions satisfying these assumptions. For instance, a $\mu > 0$ strong convexity (resp. L smoothness) parameter is a lower-bound (resp. upper-bound) on the smallest (resp. largest) eigenvalue of the hessian of f. For other classical relations, we refer to textbooks like [Bertsekas, 1997, Boyd et al., 2004, Nesterov, 2013].

1.1.4 Some Constraint Sets

Frank-Wolfe algorithms break down the minimization of a non-linear function into the multiple minimization of linear functions. Linear Minimization is an extensively studied paradigm with many results. Many practical problems come from the relaxation of combinatorial problems for which efficient linear minimization algorithms are known [Schrijver, 2003]. For example, the Birkhoff polytope is the convex hull of the permutation matrices and Linear Programming on that domain is efficiently solved with the Hungarian algorithm [Lovasz, 1986, §1.2.]. The subject is already extensively studied, and we refer to the prominent work of [Jaggi, 2013]. In particular, [Jaggi, 2013, Table 1] groups an extensive list of constraint set and associated cost of their LMO. We now review a few mechanisms.

Strongly Convex Sets. The ℓ_p balls and p-Schatten balls for $p \in [1, 2[$ are examples of strongly convex sets [Garber and Hazan, 2015]. From these, one can designed group norms which balls are also strongly convex, see [Garber and Hazan, 2015, §4] for more details. In particular, these balls admit analytical formula for the Linear Minimization Oracle for any values of the parameter p. For parameters value $p > 2$, these norm balls are more generally *uniformly convex*. In Chapter 3, we show that this more general property accelerates the Frank-Wolfe algorithm.

Atomic Sets. In Section 1.1.2, we noted that the extremal structure of the convex set $\mathcal{C}$ passes on to the (early) iterates of Frank-Wolfe algorithms. Alternatively, one can select a set of *atoms* (points) $\mathcal{A}$ sharing a specific structure and consider their convex hull $\operatorname{conv}(\mathcal{A})$ as the constraint domain in (1.1). The set of extrem points of $\operatorname{conv}(\mathcal{A})$ is a subset of $\mathcal{A}$. Hence, provided the structure of the atoms is *stable* via sparse convex combination (with is the case for instance for low-rank atoms), their structure passes on to the (early) iterates of Frank-Wolfe.

This rationale applies also in a regularization perspective, via the *gauge function* of a convex set. Loosely speaking, the gauge function allows to construct a *measurement* associated to the set $\mathcal{C}$. It is defined as follows (see [Rockafellar, 1970b, §15])

$$\Omega_{\mathcal{C}}(x) \triangleq \inf \{\lambda > 0 : x \in \lambda \mathcal{C}\}. \tag{1.8}$$

In particular, when $\mathcal{A}$ is bounded, centrally symmetric with zero its interior, the gauge of $\operatorname{conv}(\mathcal{A})$ is a norm [Rockafellar, 1970b, Theorem 15.2.]. The gauge function of the convex hull of an atomic set $\mathcal{A}$ hence defines a *regularizer* that may induce specific structures in the solution of penalized optimization problems. This is the basis to some structure inducing norms [Jacob et al., 2009, Obozinski et al., 2011, Foygel et al., 2012, Liu et al., 2012, Tomioka and Suzuki, 2013, Wimalawarne et al., 2014, Richard et al., 2014].

It is interesting to solve the atomic constraint problem with the Frank-Wolfe algorithm as the iterates (and not just the solution of (1.1)) may directly capture the atomic structure. Solving a Linear Minimization Oracle over the convex hull of an atomic domain may also be considerably cheaper that computing the proximal operator.

We finally remark in [Abernethy et al., 2018, Definition 7], the use of gauge functions as a way to define an alternative notion of a set strong-convexity. Molinaro [2020] recently prove that the two notions are equivalent.

1.2 Classic Convergence Results

We now detail known convergence rates of the Frank-Wolfe algorithm for specific structures of optimization problem (1.1). When the function is L-smooth and the domain $\mathcal{C}$ is a general compact convex set, there is a tight [Canon and Cullum, 1968, Jaggi, 2013, Lan, 2013] convergence rate of $\mathcal{O}(1/T)$.

However, accelerated convergence rates hold depending on additional structures on $\mathcal{C}$ or on the position of the solution x^* of (1.1) with respect to $\partial\mathcal{C}$. In these scenarios, the Frank-Wolfe algorithm does not require any specific knowledge of the additional structural properties and *adapts* to these scenarios. Understanding the full spectrum of structural assumptions that lead to accelerated convergence rates is thus an important research question. For instance, in Chapter 2 (resp. Chapter 3) we prove that some non-quadratic structures of the function f (resp. of the constraint set $\mathcal{C}$) accelerate the Frank-Wolfe algorithm.

This section groups convergence results that only concern the *original* Frank-Wolfe algorithm. Of course, it is not the only projection-free algorithm and the underlying open question remains as follows.

Given a problem structure, what is the best convergence acceleration a projection-free method (to be designed) can reach?

Understanding for which structures the Frank-Wolfe algorithm accelerate (and is adaptive) is important to design new projection-free algorithm. In Section 1.3, we review convergence results for the *corrective* variants of Frank-Wolfe. These are arguably designed to adapt to polyhedral domains. In this section, we also discuss results involving approximate Linear Minimization Oracle or affine invariant quantities.

Affine Invariance. The following constant curvature C_f [Clarkson, 2010a, (9)] is a measure of the non-linearity of f on the set $\mathcal{C}$.

$$C_f \triangleq \sup_{\substack{x,v\in\mathcal{C}\ \gamma\in[0,1]\\ y=x+\gamma(v-x)}} \frac{2}{\gamma^2}\Big(f(y)-f(x)-\langle\nabla f(x);y-x\rangle\Big)\ . \tag{1.9}$$

It is a key quantity in the analysis of the Frank-Wolfe algorithm [Clarkson, 2010a, Jaggi, 2013, Lacoste-Julien and Jaggi, 2013], that summarizes properties of f and the constraint set $\mathcal{C}$. It mingles together the diameter of $\mathcal{C}$ and the L-smoothness parameter. In particular, for a L-smooth function, we have $C_f \leq L\cdot\max_{x,y\in\mathcal{C}}||x-y||^2$ for any norm $||\cdot||$ [Jaggi, 2013]. It also drives the general convergence result of the Frank-Wolfe algorithm [Jaggi, 2013].

Theorem 1.2.1 (Theorem 1 in [Jaggi, 2013])**.** *When f is a L-smooth convex function and $\mathcal{C}$ a compact convex set, then the iterates of the Frank-Wolfe algorithm (with determinist line-search) satisfy*

$$f(x_T)-f(x^*)\leq\frac{C_f}{T+2}\ . \tag{1.10}$$

In [Frank and Wolfe, 1956], the sublinear convergence rate of $\mathcal{O}(1/T)$ is stated with $L\cdot\max_{x,y\in\mathcal{C}}||x-y||^2$ in place of C_f. Previous rates where not only less tight, but also depended on a specific way to measure the geometry of $\mathcal{C}$. Importantly, this affine invariant analysis with C_f (1.10), echoes to the fact that the Frank-Wolfe algorithmic procedure does not require the specification of any distance function.

The curvature constant C_f applies to other structural scenarios [Lacoste-Julien and Jaggi, 2013], but is primarily designed for the Frank-Wolfe algorithm. In Section 1.3, we review another curvature constant C_f^A [Lacoste-Julien and Jaggi, 2013], that are dedicated to *corrective* variants of Frank-Wolfe. Interestingly, in Chapter 2, we also designed quantities (error bounds) that are an interplay between the properties of the functions, the constraint domains and the types of Frank-Wolfe algorithms.

Line-Search Rules. There are two main types of line-search rules in the Frank-Wolfe algorithm beside exact line-search. The simplest one uses *oblivious* (or *determinists* as they are decided beforehand) step sizes proportional to $\frac{1}{k+1}$, where k is the number of iterations [Levitin and Polyak, 1966, Dunn and Harshbarger, 1978]. These often fail at capturing theoretical and empirical accelerated convergence rates. Alternatively, one can minimize the quadratic

upper-bound given by the L-smoothness of the function. This requires the knowledge of an upper bound on the Lipschitz constant. We call it the *short step-size* rule. Pedregosa et al. [2018] study a versatile adaptive scheme which performs short step sizes by adaptively refining an estimate of the upper bound on the Lipschitz constant L. They prove the efficiency of the approach on all known accelerated convergence regimes. Note also that [Freund and Grigas, 2016] studies constant step size accounting for warm starts.

The Frank-Wolfe algorithm enjoys accelerated convergence rates when the optimum x^* is in the interior of $\mathcal{C}$ or when the set $\mathcal{C}$ is strongly convex. In these two cases, with appropriate structural assumptions on f, the convergence rates are known to be linear. The Frank-Wolfe algorithm is *adaptive* to these scenarios as it does not need to be modified to obtain these improved convergence rates.

Acceleration I: Optimum in the interior. When the optimum is in the interior of $\mathcal{C}$ and f is a L-smooth and μ-strongly convex function, *i.e.* it enjoys additional quadratic structure, the convergence rate of the Frank-Wolfe algorithm (with exact line-search or short-step sizes) is linear [Guélat and Marcotte, 1986, Theorem 2]. It is conditioned by the non-affine invariant parameters L and μ. Lacoste-Julien and Jaggi [2013] give an affine invariant convergence result. As defined in (1.9), C_f is an affine invariance version of the L-smoothness relative to a set $\mathcal{C}$. Lacoste-Julien and Jaggi [2013] also define an affine invariant version of μ relative to $\mathcal{C}$ in the special case where x^* is in the interior of $\mathcal{C}$, see [Lacoste-Julien and Jaggi, 2013, §2] for more details.

Theorem 1.2.2 (Theorem 3 of Lacoste-Julien and Jaggi [2013])**.** *When x^* in the interior of $\mathcal{C}$, then the iterates of the Frank-Wolfe algorithm (with exact line-search or short-step-size) satisfy*

$$f(x_T) - f(x^*) \leq (1-\rho)^T (f(x_0) - f(x^*)), \tag{1.11}$$

with $\rho = min\left\{\frac{1}{2}; \frac{\mu_f^{FW}}{C_f}\right\}$, where C_f is defined in (1.9) and μ_f^{FW} in [Lacoste-Julien and Jaggi, 2013, (3)].

Note that these convergence rates depend implicitly on the distance of x^* (via μ_f^{FW}) to the boundary of $\mathcal{C}$, hence the rate can become arbitrarily bad. It however highlights that the Frank-Wolfe algorithm is adaptive to the position of the optimal solution and recovers the asymptotic convergence rate of gradient descent when the optimum is in the interior.

Acceleration II: Strongly Convex Set. When the set $\mathcal{C}$ is strongly convex (Definition 1.1.1) and there exists $c > 0$ such that $||\nabla f(x^*)|| > c$, the Frank-Wolfe algorithm (with exact line-search or short step sizes) enjoys a linear convergence rate [Levitin and Polyak, 1966, Demyanov and Rubinov, 1970]. In particular, the convergence does not require strong-convexity of the function f. In other words, the additional quadratic structure comes from the constraint set rather than from the function.

Theorem 1.2.3. *Consider $\mathcal{C}$ an α-strongly convex set with respect to a norm $||\cdot||$ and f a convex L-smooth function. Assume there exists $c > 0$ such that $inf_{x \in \mathcal{C}} ||\nabla f(x)|| > c$. The iterates of the Frank-Wolfe algorithm (with line-search of short step-sizes) satisfy*

$$f(x_T) - f(x^*) \leq (1-\rho)^T (f(x_0) - f(x^*)), \tag{1.12}$$

where $\rho = min\left\{\frac{1}{2}, \frac{\alpha c}{8L}\right\}$.

The convergence rate in (1.12) depends on $c > 0$, a measure of the minimal gradient magnitude on $\mathcal{C}$, and on the parameter α of strong convexity of the set. Both quantities depend on a specific norm and are hence are not affine invariant. To our knowledge, there exists no affine invariant analysis of the Frank-Wolfe algorithm in that setting. Such an analysis would reflect the fact that the Frank-Wolfe algorithm is *adaptive* to the scenario, with no specific input parameter depending on a choice of norm.

The two linear convergence regimes we have surveyed can both become arbitrarily bad as x^* closes the frontier of $\mathcal{C}$, and do not apply in the limit case where the unconstrained optimum lies at the boundary of $\mathcal{C}$. To this end, when the constraint set is strongly convex, Garber and Hazan [2015] prove a general sublinear rate of $\mathcal{O}(1/T^2)$ when f is L-smooth and μ-strongly convex (or slightly less than that), see [Garber and Hazan, 2015, Theorem 2].

The Analysis of Dunn. In [Dunn, 1979, 1980], Dunn proves accelerated (linear) convergence rates of the Frank-Wolfe algorithm when the optimization problem has a sufficient *quadratic structure* at $x^* \in \partial\mathcal{C}$. In particular, his convergence results non-trivially subsume the cases where $\mathcal{C}$ is globally or locally strongly convex. In Figure 1.1, we illustrate scenarios where the set is locally not strongly-convex, but the Frank-Wolfe algorithm still enjoys linear convergence. Geometrically, it is sufficient for the linear convergence of the Frank-Wolfe algorithm that there exists a tangent hyperball at the solution $x^* \in \partial\mathcal{C}$ with the (non-zero) gradient normal to this hyperball at x^*. Algebraically, for $x^* \in \partial\mathcal{C}$, Dunn [1979] introduces the following quantity

$$a_{x^*}(\sigma) = \inf_{\substack{x \in \mathcal{C} \\ ||x - x^*|| \geq \sigma}} \langle \nabla f(x^*); x - x^* \rangle. \tag{1.13}$$

In [Dunn, 1979], lower-bounds on $a(\sigma)$ determine the converge rate of the Frank-Wolfe algorithm. In particular, when there exists $A > 0$ such that $a_{x^*}(\sigma) \geq A\sigma^2$, the Frank-Wolfe algorithm (with exact line-search or short step-sizes) converges linearly.

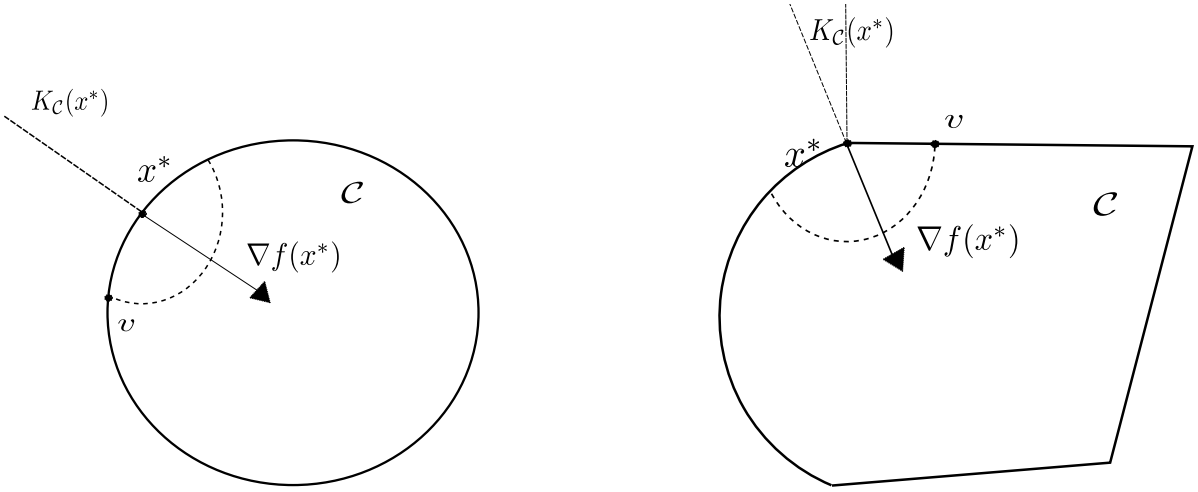
Approximate LMO. It is also possible to approximatively solve the Linear Minimization Oracle (line 2 to Algorithm 9) while maintaining the convergence guarantees. This is for instance useful in situations where solving the exact Linear Minimization Oracle is an intractable problem, but efficient approximate solutions exist. There are several ways to quantify the approximation: *multiplicatively* (see [Lacoste-Julien et al., 2013, Appendix C]) or *additively* (see [Dunn and Harshbarger, 1978, Jaggi, 2013] or [Freund and Grigas, 2016, Section 5]). An *additive* δ-approximate solution $\tilde{v}$ to the Linear Minimization Oracle for a convex set $\mathcal{C}$ at iterate x_t satisfies

$$\langle -\nabla f(x_t); \tilde{v} \rangle \geq \max_{v \in \mathcal{C}} \langle -\nabla f(x_t); v \rangle - \delta. \tag{1.14}$$

For $\nu \in [0, 1]$, a *multiplicative* ν-approximate solution $\tilde{v}$ to the Linear Minimization Oracle for a convex set $\mathcal{C}$ for $d \in \mathbb{R}^d$ satisfies

$$\langle -\nabla f(x_t); \tilde{v} - x_t \rangle \geq \eta \cdot \max_{v \in \mathcal{C}} \langle -\nabla f(x_t); v - x_t \rangle. \tag{1.15}$$

It is then in case-by-case basis that extensions of Frank-Wolfe admit convergence results when using approximate Linear Minimization Oracles. In Lemma 1.A.1 of Appendix 1.A,



(a) $a_{x^*}(\sigma) \geq A||\nabla f(x^*)||\sigma^2$ (b) $a_{x^*}(\sigma) \geq A||\nabla f(x^*)||\sigma^2$

Figure 1.1: In both cases, there exists $A > 0$ such that $a_{x^*}(\sigma) \geq A||\nabla f(x^*)||\sigma^2$, and the analysis of Dunn guarantees linear convergence rate. Note however, that on the right figure, $\mathcal{C}$ is not locally strongly-convex at x^*. The analysis of Dunn goes beyond local strong-convexity. Note that in the right figure, at x^*, there is a quadratic lower bound on $a_{x^*}(\sigma)$ as soon as the negative gradient $-\nabla f(x^*) \in K_{\mathcal{C}}(x^*)$ is not orthogonal to the face of $\mathcal{C}$ that contains v. The dashed circled represents $\mathcal{C} \cap \{x \mid ||x - x^*|| = \sigma\}$.

we provide a proof of the linear convergence of the Frank-Wolfe algorithm when the set in a strongly convex set and using multiplicative approximate LMO. Note that [Pedregosa et al., 2018] studied approximate LMO for *corrective* variants of Frank-Wolfe.

1.3 Corrective Frank-Wolfe Algorithms

We now introduce the *corrective* or *away* versions of the Frank-Wolfe algorithm. These are designed to use projection-free (linear minimization) oracles that maintain the optimization iterates in the convex feasible region $\mathcal{C}$. They introduce additional type of descent directions with respect to the original Frank-Wolfe algorithm.

They are called *corrective* when considering the iterates from an algebraical point of view: these directions allow to *correct* carefully chosen weights of the current (sparse) convex combination of the iterate x_t. Alternatively, these additional directions are also called *away* or *in-face* directions when considering the algorithm from a geometrical point of view. Indeed, these directions move the current iterate in the current face it belongs to, or *away* from some selected vertices in the iterates convex combination.

There are many such variants that explore different trade-off and algorithmic designs: various criterions to chose between a classic Frank-Wolfe direction or an *away* direction; the type of *away* direction; designing non-atomic versions and many others. Importantly, these methods allow to *adaptively* capture linear convergence rates when the constraint set is a polytope and the objective function is L-smooth and μ-strongly convex.

1.3.1 Away or Corrective Mecanisms

Away-steps Frank-Wolfe. It was first proposed in [Wolfe, 1970] and analyzed in [Guélat and Marcotte, 1986]. Along with the iterates (x_t), the algorithm maintains the point set $\mathcal{S}_t$ and the sequence of weights $(\alpha_v)_{v\in\mathcal{S}_t}$ such that $x_t = \sum_{v\in\mathcal{S}_t} \alpha_v v$. The points in $\mathcal{S}_t$ are extreme points of $\mathcal{C}$. The main insight is that for any $v \in \mathcal{S}_t$, $x_t - v$ updates of the form $x_{t+1} = x_t + \gamma(x_t - v)$ with $\gamma \in [0, \alpha_v/(1-\alpha_v)]$ maintains $x_{t+1} \in \mathcal{C}$. These hence define feasible directions that do not require any projection step.

In line 5, the algorithm then selects the vertex $v \in \mathcal{S}_t$ such that $x_t - v$ is the best possible descent direction, *i.e.* the direction most correlated with the negative gradient. This vertex is called the *away* vertex. In line 7, the algorithm then arbitrates between the Frank-Wolfe and the away direction. It chooses the one most correlated with the negative gradient. Note that many works later considered different choosing criterion.

Algorithm 2 Away-steps Frank-Wolfe (AFW)

Input: $x_0 \in \mathcal{C}$, $x_0 = \sum_{v\in\mathcal{S}_0} \alpha_v^{(0)} v$ with $|\mathcal{S}_0| = s$.

1:
2: **for** $t = 0, 1 \dots, T$ **do**
3: Compute $v_t^{\mathrm{FW}} = \mathrm{LMO}_{\mathcal{C}}(\nabla f(x_t))$
4: Let $d_t^{\mathrm{FW}} = v_t^{\mathrm{FW}} x_t$ ▷ FW direction
5: Compute $v_t^A = \mathrm{LMO}_{\mathcal{S}_t}(-\nabla f(x_t))$
6: Let $d_t^A = x_t - v_t^A$. ▷ Away direction
7: **if** $\langle -\nabla f(x_t), d_t^{\mathrm{FW}} \rangle \geq \langle -\nabla f(x_t), d_t^A \rangle$ **then**
8: $d_t = d_t^{\mathrm{FW}}$ and $\gamma_{\max} = 1$ ▷ FW step
9: **else**
10: $d_t = d_t^A$ and $\gamma_{\max} = \alpha_{v_t}^{(t)}/(1-\alpha_{v_t}^{(t)})$ ▷ Away step
11: **end if**
12: Set γ_t by line-search, with $\gamma_t = \mathrm{argmax}_{\gamma\in[0,\gamma_{\max}]} f(x_t + \gamma d_t)$
13: Let $x_{t+1} = x_t + \gamma_t d_t$ ▷ update $\alpha^{(t+1)}$
14: Let $\mathcal{S}_{t+1} = \{v \in \mathcal{A} \text{ s.t. } \alpha_v^{(t+1)} > 0\}$
15: **end for**

Output:

Algorithm 2 is a *corrective* version of the Frank-Wolfe algorithm because some iterations explicitly *correct* one of the weights of the iterate convex combination. This might appear as a loose statement as any iteration of the original Frank-Wolfe algorithm also multiplicatively rescales the weights by $(1-\gamma_t)$, where γ_t is the step-size. The difference is that Frank-Wolfe iterations correct the representation by adding new weights and rescaling accordingly. Away-steps certainly only correct the current convex combination of the iterates. Note that some other versions, like pair-wise Frank-Wolfe (see Section 1.3.3) are designed to modify only two weights at each iteration.

Fully-Corrective Frank-Wolfe. The away-step Frank-Wolfe focuses primarily on finding a feasible direction that best correlates with the negative gradient. The *corrective* property of the away-steps appears only as a by-product of that quest. Other trade-offs between *correcting* the iterates convex representations and obtaining the best immediate local primal decrease can be

considered. In particular, the fully corrective Frank-Wolfe [Von Hohenbalken, 1977, Holloway, 1974, Hearn et al., 1987] can be viewed as being on the other side of the checkboard. Before seeking for immediate local primal decrease via a Frank-Wolfe direction, it searches for the best point (in primal decrease) in the convex hull of the points appearing in the decomposition of the current iterate, a *correction* of the current iterate, see line 4 in Algorithm 3.

Algorithm 3 Fully-Corrective Frank-Wolfe (FCFW)

Input: $x_0 \in \mathcal{C}$, $x_0 = \sum_{v \in \mathcal{S}_0} \alpha_v^{(0)} v$ with $|\mathcal{S}_0| = s$.

1:
2: **for** $t = 0, 1 \dots, T$ **do**
3: Compute $v_t = \text{LMO}_{\mathcal{C}}(\nabla f(x_t))$
4: $(x_{t+1}, \mathcal{S}_{t+1}) = \text{argmin}_{x \in \text{Conv}(\mathcal{S}_t \bigcup \{v_t\})} f(x)$
5: **end for**

Output:

1.3.2 Linear Convergence on Polytopes

The lower bounds on the Frank-Wolfe algorithm show that in generality, without algorithmic modification, the algorithm could not converge at a linear rate when the function is smooth, enjoys *quadratic structures* (like strong-convexity) and the set is a polytope. In particular, linear convergence results on the Frank-Wolfe algorithm are known only when the solution of (1.1) is in the interior, or when the set also has *quadratic structure* [Levitin and Polyak, 1966, Demyanov and Rubinov, 1970, Dunn, 1979] – see Section 1.2. The *corrective* versions of Frank-Wolfe were in particular designed to alleviate these issues.

Recently, two bodies of works [Lacoste-Julien and Jaggi, 2013, Garber and Hazan, 2013a, Lacoste-Julien and Jaggi, 2015b] showed with different techniques that indeed some versions of Frank-Wolfe enjoy global linear convergence rate when the set is a polytope – and the function f has adequate structure. Note also that Beck and Shtern [2017] prove linear convergence under a quadratic error bound instead of the strong-convexity of f, which is a localized *quadratic structure* on f.

Lacoste-Julien and Jaggi [2013, 2015b] give an affine invariant linear convergence result for *corrective* variants of Frank-Wolfe. Garber and Hazan [2013a,b] also exhibit a modification of the Frank-Wolfe algorithm enjoying linear convergence. Their algorithm relies on a modification of the Frank-Wolfe algorithm, where *Local Linear Minimization Oracles* (LLMO) replace the Linear Minimization Oracles. This new oracle is efficiently performed when the constraint set is a polytope, although it requires some function parameters. The LLMO is a *relaxation* of a stronger oracle that minimizes a linear function over the intersection of the original set $\mathcal{C}$ and a ball resulting from the strong convexity of f. Lan [2013] considers a version of Frank-Wolfe [Lan, 2013, Algorithm 3] with this (expensive) *enhanced* LMO that admits linear minimization oracle. Interestingly, the link between Frank-Wolfe with LLMO and its *corrective* variants is not straightforward from the implementation of the LLMO for polytopes.

Under these assumptions, the seek for linear convergence proofs of a modified Frank-Wolfe algorithm has a long history. [Guélat and Marcotte, 1986] gave the first linear convergence proof with a strict complementarity assumption, *i.e.* when the constrained optimum is in the relative interior of its optimal face, and the unconstrained optimum is away from the boundary.

Beck and Teboulle [2004] show linear convergence of the Frank-Wolfe algorithm under a slater condition on their original problem. It is very close to assuming that the optimum is in the relative interior of the constraint set $\mathcal{C}$. Without restriction on the position of the optimum, Migdalas [1994], Lan [2013] gave linear convergence (for $\mathcal{C}$ polytope and f smooth and strongly-convex) rates but with much stronger oracles that are akin to projection or proximal steps. Todd and Yıldırım [2007] prove linear convergence when $\mathcal{C}$ is the simplex and with no precise dimension dependency of the conditioning number; [Damla Ahipasaoglu et al., 2008, Kumar and Yıldırım, 2011] assume the Robinson Condition [Robinson, 1982].

Affine Invariance. As for the Frank-Wolfe algorithm, Lacoste-Julien and Jaggi [2013] proposes an affine invariant notion of the L-smoothness that is dedicated to the analysis of *corrective* variants of Frank-Wolfe. We recall the definition of *away curvature* in [Lacoste-Julien and Jaggi, 2015a, Appendix D], with

$$C_f^A \triangleq \sup_{\substack{x,s,v\in\mathcal{C}\\ \eta\in[0,1]\\ y=x+\eta(s-v)}} \frac{2}{\eta^2}\big(f(y)-f(x)-\eta\langle\nabla f(x), s-v\rangle\big), \tag{1.16}$$

where f and $\mathcal{C}$ are defined in problem (2.4) above.

Convergence Rates Conditioning. Many alternatives to the original *Pyramidal Width* of [Lacoste-Julien and Jaggi, 2013], which conditions the linear convergence proof of their analysis, have been considered. The conditioning in [Garber and Hazan, 2013a, Theorem 2] directly depends on problem parameters such as L-smooth, μ-strongly convex parameters or the ambient dimension but [Garber and Hazan, 2013a, Algorithm 2] depends on such specific parameters. This echoes the stronger versions of [Lan, 2013] where the theoretical convergence rates conditioning is also explicit in these parameters. The works of [Garber and Meshi, 2016, Bashiri and Zhang, 2017] proposed a condition number that depends on the dimension on the optimal face on which the solution of (1.1) lies, which may be considerably smaller than the ambient dimension. Beck and Shtern [2017] proposes a *vertex-facet distance constant* condition number. Finally, [Pena and Rodriguez, 2018, Gutman and Pena, 2018] studied various interesting geometrical notions of conditioning relative to a polytope.

1.3.3 Other Corrective Variants

Many other *corrective* variants of Frank-Wolfe exist. These rely on different ways to choose between Frank-Wolfe direction and *corrective* direction or to implement the Fully-Corrective Oracle. For instance [Vinyes and Obozinski, 2017] propose an efficient version of Fully-Corrective Frank-Wolfe dedicated to difficult atomic sets and where the *corrective* oracle relies on a specific active-set algorithm. Some also design new possible projection-free *corrective* directions or construct non-atomic based algorithmic versions.

Pair-Wise Frank-Wolfe. The Pair-Wise Frank-Wolfe algorithm [Mitchell et al., 1974], was revisited in [Lacoste-Julien and Jaggi, 2015a]. Contrary to the AFW (Algorithm 2), it considers only one type of descent direction, parallel to the line joining the Frank-Wolfe vertex (line 3 of Algorithm 2) and the Away vertex (line 5 of Algorithm 2). Interestingly, at each iteration,

this algorithm makes a convex update that *corrects* exactly two weights of the current decomposition of the iterate. Its main drawback is not practical but theoretical as it becomes much harder to account for the number of *drop steps* and hence influence the linear convergence guarantees, see [Lacoste-Julien and Jaggi, 2015a].

Algorithm 4 Pair-Wise Frank-Wolfe (PFW)

Input: $x_0 \in \mathcal{C}$, $x_0 = \sum_{v \in \mathcal{S}_0} \alpha_v^{(0)} v$ with $|\mathcal{S}_0| = s$.

1:
2: **for** $t = 0, 1 \ldots, T$ **do**
3: Compute $v_t^{\text{FW}} = \text{LMO}_{\mathcal{C}}(\nabla f(x_t))$ and $v_t^A = \text{LMO}_{\mathcal{S}_t}(-\nabla f(x_t))$
4: Let $d_t = v_t^{\text{FW}} - v_t^A$. ▷ Pair-wise direction
5: Set γ_t by line-search, with $\gamma_t = \text{argmax}_{\gamma \in [0, \alpha_{v_t^A}]} f(x_t + \gamma d_t)$
6: Let $x_{t+1} = x_t + \gamma_t d_t$ ▷ update $\alpha^{(t+1)}$ (see text)
7: Update $\mathcal{S}_{t+1} = \{v \in \mathcal{A} \text{ s.t. } \alpha_v^{(t+1)} > 0\}$
8: **end for**

Output:

Min-Norm Point. Min-Norm Point (MNP) algorithm [Wolfe, 1976] is also known to be a *corrective* variant of Frank-Wolfe algorithms. Note also that [Bach et al., 2013, §9.2.] pointed out that the min-norm point is a particular instance of *the Active-Set Method for QP* in [Nocedal and Wright, 2006, Algorithm 16.3 in Chapter 16.5.] when the hessian equal to the identity. Min-Norm Point relies on a sequence of affine projections.

Forward-Backward. The forward-backward method of Rao et al. [2015] is a specific way of performing the fully-corrective step in Algorithm 3, where the *forward* steps correspond to Frank-Wolfe steps and the *backward* steps correspond to *corrective* steps in Algorithm 3 and 2.

Memory-Less Corrective Versions of Frank-Wolfe. *Away* or *corrective* versions of Frank-Wolfe as detailed in §1.3.1 perform an additional Linear Minimization Oracle than the Frank-Wolfe algorithm. This LMO requires to store the optimization iterates as convex combinations of atoms. When these algorithms are used to leverage on the projection-free property of the Frank-Wolfe framework – and not necessarily on the trade-off between the structure of the iterates and their approximation quality –, they suffer from a memory overhead, with respect to the Frank-Wolfe Algorithm. There is also a possible runtime overhead because the selection of the best away vertex relies on an enumeration strategy.

For a specific class of polytopes – *i.e.* polytopes with vertices in $\{0, 1\}^d$ and for which we have access to an algebraic representation –, [Garber and Meshi, 2016] first showed that it is possible to compute an away-step without relying on a specific decomposition of the current iterate. They show that their decomposition invariant version of Frank-Wolfe enjoy linear convergence rates on these polytopes when the function is L-smooth and μ-strongly convex. In particular, they first exhibit conditioning that depends on the sparsity of the optimal solution in term of vertices of the polytopes, *i.e.* on the dimension of the optimal face. Their work was then extended and refined by [Bashiri and Zhang, 2017] to general polytopes.

In-Face Frank-Wolfe. [Freund et al., 2017] propose an *In-face* version of Frank-Wolfe algorithms with several key features for the specific application of matrix completion. *In-Face* refers to the fact that *away* direction as developped in Section 1.3.1 are directions in the affine hull of the optimal face the current iterate belongs to. Leveraging on the specific structures of the nuclear ball [So, 1990], [Freund et al., 2017] hence proposed *corrective* directions that depends on the optimal face $\mathcal{F}_{\mathcal{C}}(x_t)$ the current iterate x_t belongs to. Hence, similarly to [Garber and Meshi, 2016, Bashiri and Zhang, 2017], it does not rely on a non-affine invariant atomic representation of the current iterate.

Moreover, they propose a different choosing criterion between classical Frank-Wolfe directions and these *in-face* directions. In particular, these chosen directions are more favourable to *in-face* directions which empirically results in a sparser trade-off between accuracy and structure of the iterate. In the case of matrix completion, this corresponds to the trade-off between data fidelity and low-rank structure of the solution. One can loosely interpret these criteria as practical algorithmic schemes between the classic away version of Frank-Wolfe and fully-corrective versions.

1.4 Applications and Variants

In the previous section, we presented the original Frank-Wolfe algorithm, its *corrective* variants and the known scenarios where these algorithms enjoy accelerated convergence rates. We now present some of the many different mechanisms that can be plugged in these algorithms to account for the various specificities of practical applications. For instance, there are stochastic, block-coordinate [Lacoste-Julien et al., 2013, Osokin et al., 2016], second-order [Carderera and Pokutta, 2020], non-convex [Dunn, 1980, Lacoste-Julien, 2016], non-smooth and many other different versions of Frank-Wolfe algorithms. In Section 1.4.1, we review some of these mechanisms and in Section 1.4.2 we point to applications leveraging them.

1.4.1 Other Mechanisms

Stochastic Frank-Wolfe. In many practical scenarios, stochastic versions of the gradient are easily accessible and computationally cheaper than the exact gradients. This is, for instance, the case in Empirical Risk Minimization. The function f in (1.1) is the sum of n functions, where n is the size of the dataset, and computing an exact gradient requires the computation of the gradient of n function. It is often preferable to compute less expensive stochastic versions of this gradient over a random *batch* of these n functions. Algorithms using such stochastic estimators of the gradient are usually called *stochastic*. Some effort has been dedicated to designing stochastic Frank-Wolfe algorithms, with versions increasing the batch-size at each iteration [Hazan and Luo, 2016, Reddi et al., 2016] and other recently converging with fixed batch-sizes [Mokhtari et al., 2018, Hassani et al., 2019, Zhang et al., 2019, Lu and Freund, 2020] and enjoying fast convergence rates [Négiar et al., 2020].

Reducing Number of Call to Oracles. Some works focused on producing Frank-Wolfe algorithms with the same asymptotic convergence guarantees as their original counter-part while requiring less gradient or LMO calls. For instance, some versions leverage on the *gradient sliding* mechanism [Lan and Zhou, 2014], where the gradient computation are recycled at subsequent iterations [Cheung and Lou, 2015, Lan and Zhou, 2016, Qu et al., 2017]. Also,

Braun et al. [2017b] propose a *lazy* mechanism to reduce the number of calls to the *full* Linear Minimization Oracle. A *weak separation* oracle (that is much weaker than an approximate Linear Minimization Oracle) replaces full LMO calls when these are unnecessary (see also their stochastic version in [Lan et al., 2017].

Affine Spaces and Matching-Pursuit. Recent works have shown the relation between Frank-Wolfe algorithms and Matching-Pursuit algorithms [Locatello et al., 2017b], which also relies on linear minimization oracle over affine sets. While previous work also considered affine directions [Wolfe, 1976], this connection has been particularly fruitful [Locatello et al., 2017c, Combettes and Pokutta, 2020].

Cone Constrained or Non-Negative Matching-Pursuit. For cone constrained problem (or non-negative matching-pursuit algorithms), [Locatello et al., 2017c] designed the first projection-free algorithms with convergence guarantees for general L-smooth convex functions enjoying sublinear and linear rates. In particular they overcome the difficulty that when the constrained set in a conic hull of an atomic set, – as opposed to the convex or affine hull of an atomic set – classical generalization of MP to non-negative constraints do not satisfy the *alignement property*, see [Locatello et al., 2017c, §2] and [Pena and Rodriguez, 2018]. This property states that at any suboptimal iterates, there exists a search direction given by the algorithm that is non-negatively correlated with the negative gradient. Hence [Locatello et al., 2017c, Algorithm 2] proposes $-\frac{x_t}{||x_t||}$ as a possible feasible direction that guarantees that, unless at the optimum, the algorithm always picks a direction striclty positively correlated with the negative gradient. Informally, the first algorithm they propose can be seen as the equivalent of plain Frank-Wolfe algorithm on a cone constrained setting in the sense that it is a first-order projection-free algorithm that suffers from a general sublinear rate of $\mathcal{O}(1/T)$. Indeed when the algorithm chooses the direction $-\frac{x_t}{||x_t||}$, it changes all the weights. Hence they propose *corrective* versions [Locatello et al., 2017c, Algorithm 3 and 4] that are the analogous of the *away* or *corrective* versions of Frank-Wolfe we reviewed in §1.3. In particular, under appropriate assumption, these enjoy linear convergence rates with a similar analysis as in [Lacoste-Julien and Jaggi, 2013].

Generalized Frank-Wolfe. In this dissertation, we focus on designing and analyzing versions of Frank-Wolfe algorithms for constrained optimization problems. However, conditional gradient methods have also been studied and designed for different minimization problem formulations. There are *generalized* Frank-Wolfe algorithms for penalized problems or composite problems Bredies et al. [2009], Dudik et al. [2012], Harchaoui et al. [2012], Vinyes and Obozinski [2017] of the form

$$\min_{x \in K} f(x) + \Phi(x), \tag{1.17}$$

where K is a cone and Φ a penalty function. These types of Frank-Wolfe algorithms are arguably called *generalized* because the optimization iterates are unconstrained. The iterates do not necessarily evolve in the constrained sets of the equivalent constraint formulation of the problem. Often, the function Φ is chosen to be the gauge of the convex hull of an atomic set [Dudik et al., 2012, Harchaoui et al., 2012, Vinyes and Obozinski, 2017] (see Section 1.1.4), or more generally a gauge-like function [Rockafellar, 1970b, Theorem 15.3.], with an emphasis on

the composition of a non-decreasing function with an atomic gauge function [Harchaoui et al., 2015, Sun and Bach, 2020].

These algorithms offer another perspective with respect to the constrained versions and were studied theoretically in [Harchaoui et al., 2015, Bach, 2015, Yu et al., 2017, Nesterov, 2018]. Interesting connections have been made with other well-known algorithms like iterative shrinkage method [Bredies et al., 2009], mirror descent [Bach, 2015] or column generation algorithm [Vinyes and Obozinski, 2017].

Online Learning. There is a strong interplay between projection-free online learning, linear online learning and Frank-Wolfe algorithms for offline optimization. New algorithms and convergence analysis have emerged from the interplay.

For instance, Hazan and Kale [2012] first propose a projection-free (*i.e.* one linear minimization per iteration) online algorithm that can be seen as a direct transposition of the Frank-Wolfe algorithm in an online setting. The general $\mathcal{O}(1/T)$ convergence rate for smooth function translate into a $\tilde{\mathcal{O}}(\sqrt{T})$ regret bound of Online Frank-Wolfe [Hazan and Kale, 2012, Algorithm 1] in the stochastic setting. Similarly Garber and Hazan [2013a] first proposed a projection-free online algorithm that obtain a logarithmic $\tilde{\mathcal{O}}(\log(T))$ regret bound when the decision set is a polytope and the cost functions are smooth and strongly convex. Lafond et al. [2015] also transpose the *away* algorithms of [Lacoste-Julien and Jaggi, 2013, 2015b] and their analysis to the online setting. Previous work to Hazan and Kale [2012] seem to have consider only online linear optimization [Kalai and Vempala, 2005, Huang et al., 2016a]. The Frank-Wolfe setting hence provide a principled efficient manner to transform non-linear problems into a series of linear steps.

Alternatively, the work of [Abernethy and Wang, 2017, Abernethy et al., 2018] explore the other way around and derive some Frank-Wolfe algorithms by opposing two online algorithms. They derive new FW algorithms. For instance, [Abernethy et al., 2018, Algorithm 2] has the same convergence rate of $\mathcal{O}(1/T^2)$ when the set and function are strongly convex [Abernethy et al., 2018, corollary 11].

1.4.2 Examples of Applications of Frank-Wolfe

Frank-Wolfe algorithms appear in a wealth of applications, like SVM [Clarkson et al., 2012, Ñanculef et al., 2014, Osokin et al., 2016], submodular optimization [Edmonds, 2003, Bach et al., 2013, Bian et al., 2016, Hassani et al., 2017, Mokhtari et al., 2017], coresets [Kumar and Yildirim, 2005, Damla Ahipasaoglu et al., 2008, Clarkson, 2010a], neural network pruning [Ping et al., 2016, Scardapane et al., 2017], optimal control theory [Kelley, 1962, Gilbert, 1966, Barnes, 1972, Dunn, 1974, Kumar, 1976, Dunn, 1979, 1980], matrix completion [Shalev-Shwartz et al., 2011, Harchaoui et al., 2012, Dudik et al., 2012, Giesen et al., 2012, Allen-Zhu et al., 2017, Yurtsever et al., 2017] and many others. Let us now regroup a non-exhaustive list of applications of the Frank-Wolfe algorithms.

Non-Hilbert Spaces. Frank-Wolfe methods have also been leveraged as an appealing solution in non-Hilbert spaces. Indeed, each iteration relies on feasible directions that only involve the current iterate and an extreme point of the constraint set. In particular, it was considered for measure spaces [Bredies and Pikkarainen, 2013, Boyd et al., 2017, Denoyelle et al., 2019, Luise et al., 2019]. In that context only non-corrective variants of the Frank-Wolfe method

are leveraged. Indeed the *corrective* variants are known to accelerate convergence when the constraint set is a polytope, which is of limited interest when the underlying space is infinite dimensional. It was also extensively used in optimal transport applications [Courty et al., 2016, Vayer et al., 2018, Paty and Cuturi, 2019, Luise et al., 2019].

Herding. [Bach et al., 2012] recently leveraged on the various algorithmic versions and analysis of conditional gradient algorithms to find good quadratic rules to approximate integrals in Reproducing Kernel Hilbert Spaces (RKHS) with norm $||\cdot||_{\mathcal{H}}$, see [Lacoste-Julien et al., 2015, §2.1.] for a complete introduction. Let $\mathcal{X}$ be the data point space, Φ the map from $\mathcal{X}$ to the RKHS and p a fixed distribution on $\mathcal{X}$. Bach et al. [2012] proposed solving with Frank-Wolfe Algorithms the following problem

$$\operatorname*{argmin}_{g \in \mathcal{M}} ||\mu_p - g||, \tag{1.18}$$

where $\mathcal{M} \triangleq \text{Conv}(\Phi(x) \mid x \in \mathcal{X})$ is known as the *marginal polytope* and $\mu_p \triangleq \mathbb{E}_p\Big(\Phi(x)\Big)$ is the *mean element*, see [Lacoste-Julien et al., 2015, §2.1.] for the rational behind (1.18). Using various Frank-Wolfe algorithms to solve (1.18) gives differently weighted iterates $g_t = \sum_{t=1}^{T} w_t e_t$ where e_t are extreme points of the marginal polytope $\mathcal{M}$ which actually are of the form $\Phi(x)$ under some mild assumptions. Hence $g_t = \sum_{t=1}^{T} w_t \Phi(x_t)$, which can be identified to the quadratic rule $\tilde{p_t} \triangleq \sum_{i=t}^{T} w_t \delta_{x_t}$.

This was the basis of improvments for Kernel Herding [Bach et al., 2012], particle filtering [Lacoste-Julien et al., 2015], MMD [Futami et al., 2019] or Bayesian Inference [Belanger et al., 2013, Niculae et al., 2018].

Computer-Vision. In structured prediction, the relations between objects are modelled via hard constraints. Recent computer vision applications involve large scale settings. Frank-Wolfe methods have been leveraged to deal with constrained discriminative clustering like in action localization [Bojanowski et al., 2014], text-to-video alignment [Bojanowski et al., 2015, Alayrac et al., 2016], object co-localization in videos and images [Joulin et al., 2014] or instance-level segmentation [Seguin et al., 2016]. In these particular cases, the domains are sometimes products of simpler domains. Hence block-coordinate Frank-Wolfe methods can be used to scale the problems [Miech et al., 2017, Peyre et al., 2017, Miech et al., 2018].

Appendices

1.A Proofs

1.A.1 More on Approximate LMO

Here we provide a version of the linear convergence of the Frank-Wolfe algorithm with approximate Linear Minimization Oracle when the set is strongly convex. It is not much different from the proof with exact Linear Minimization Oracle, but we could not find any reference for it. Also, the dependence of the bound on the error is not completely favorable as it depends on the square of the multiplicative approximation error parameter.

Lemma 1.A.1 (FW on Strongly Convex Set with Approximate Oracle)**.** *Assume f is a convex L-smooth function, $\mathcal{C}$ an α-strongly convex set and $\inf_{x\in\mathcal{C}}||\nabla f(x)|| > c > 0$. Assume the LMO in the Frank-Wolfe algorithm (line 2 in Algorithm 9) is solved with a multiplicative error $\eta \in [0,1]$, i.e. the Frank-Wolfe vertex v_t satisfies*

$$\langle -\nabla f(x_t); v_t - x_t\rangle \geq \eta \cdot \max_{v\in\mathcal{C}} \langle -\nabla f(x_t); v - x_t\rangle \ .$$

Then the Frank-Wolfe iterates (with short-step size or exact line-search) satisfies

$$f(x_K) - f(x^*) \leq \rho^K (f(x_0) - f(x^*)), \tag{1.19}$$

with $x_0 \in \mathcal{C}$ and $\rho \triangleq max\Big\{\frac{\eta}{2}; 1 - \frac{c\eta^2\alpha}{4L}\Big\}$.

Proof. By L-smoothness and choice of line search, for any $\gamma \in [0,1]$, we have

$$f(x_{t+1}) \leq f(x_t) + \gamma\langle \nabla f(x_t); v_t - x_t\rangle + \frac{\gamma^2}{2} L||v_t - x_t||^2 \ .$$

By strong convexity of $\mathcal{C}$, $\tilde{v}_t(z) = \frac{v_t+x_t}{2} + \frac{\alpha}{4}||v_t - x_t||^2 z$ belong to $\mathcal{C}$ for any unit vector z. Because v_t is an η-multiplicative approximate LMO, we have (recall $h_t = f(x_t) - f(x^*)$)

$$\begin{aligned}
\langle \nabla f(x_t); v_t - x_t\rangle &\leq \eta\langle -\nabla f(x_t); \tilde{v}_t(z) - x_t\rangle \\
\langle \nabla f(x_t); v_t - x_t\rangle &\leq \frac{\eta}{2}\langle -\nabla f(x_t); v_t - x_t\rangle + \frac{\eta\alpha}{4}||v_t - x_t||^2\langle -\nabla f(x_t); z\rangle \\
\langle \nabla f(x_t); v_t - x_t\rangle &\leq -\frac{\eta}{2} h_t - \frac{\eta\alpha}{4}||v_t - x_t||^2 c
\end{aligned}$$

So finally

$$h_{t+1} \leq h_t(1 - \frac{\eta\gamma}{2}) + ||v_t - x_t||^2\Big(\frac{\gamma^2}{2}L - \frac{c\eta\alpha\gamma}{4}\Big) \ .$$

Hence if $\frac{c\eta\alpha}{2L} \geq 1$, we set $\gamma = 1$ and we have $h_{t+1} \leq h_t(1 - \frac{\eta}{2})$. Otherwise we chose $\gamma = \frac{c\eta\alpha}{2L}$ and we have

$$h_{t+1} \leq h_t(1 - \frac{c\eta^2\alpha}{4L}).$$

Finally we have,

$$h_{t+1} \leq h_t \cdot \max\Big\{\frac{\eta}{2}; 1 - \frac{c\eta^2\alpha}{4L}\Big\}. \tag{1.20}$$

■

Chapter 2

Restarting Frank-Wolfe

In this chapter we consider constrained convex minimization problems of the form

$$\min_{x \in \mathcal{C}} f(x),$$

where f is a smooth convex function and $\mathcal{C}$ is a compact convex set. Our goal is to adapt and analyze new versions of the Frank-Wolfe algorithms which enjoy accelerated convergence rates under specific conditions. The results in this chapter contribute to suggesting that the Frank-Wolfe algorithms are adaptive to various type of structures of the objective function.

We replace μ-strongly convex assumptions with specially designed error bounds type conditions. We analyze a restarted version of the away-steps Frank-Wolfe when the set is a polytope and a restarted version of Frank-Wolfe when the optimum is in the interior of $\mathcal{C}$. Our results fill the gap between the previous linear $\mathcal{O}(\log 1/\epsilon)$ rate and the sublinear $\mathcal{O}(1/\epsilon)$ rate. Our contributions can be summarized as follows.

1. *Strong-Wolfe primal bound.* Under generic assumptions, we derive strong-Wolfe primal gap bounds generalizing those obtained from strong convexity of f. These bounds are obtained by combining a Łojasiewicz growth condition on f with a scaling inequality on $\mathcal{C}$, and continuously interpolate between the convex and strongly convex cases. In particular, they can be considered as a type of first-order error bounds designed for Frank-Wolfe algorithms.

2. *Fractional Frank-Wolfe Algorithms.* We then define a new conditional gradients algorithm that dynamically adapts to the parameters of these strong-Wolfe primal bounds using a restart scheme. The resulting algorithm achieves either sub-linear (i.e., $\mathcal{O}(1/\epsilon^q)$ with $q \leq 1$) or linear convergence rates depending on the strong-Wolfe primal gap parameters. The exponent q depends on the growth of the function around the optimum, so the function is not required to be strongly convex in the traditional sense. In particular, we obtain linear rates (depending on the parameters) for non-strongly convex functions. Our rates are satisfied after a mild burn-in phase that does not depend on the target accuracy.

3. *Robust restarts.* Restart schedules often heavily depend on the value of unknown parameters. We show that because the Frank-Wolfe methods naturally produce a stopping criterion in the form of the strong-Wolfe gap, our restart schemes are robust and do not require knowledge of the unobserved strong-Wolfe primal gap bound parameters.

4. We generalize our approach to Hölder smooth functions.

In Section 2.1 we quickly review error bounds conditions and their applications in first-order optimization algorithms. In Section 2.2 we briefly recall key notions, notations and we then describe our strong-Wolfe primal bounds. In Section 2.3 we present the Fractional Away-step Frank-Wolfe algorithm along with the associated restart schemes in Section 2.4. Section 2.5 gives an analysis when the optimum is in the interior of the constraint set (see Section 2.5.2). For completeness, in Appendix 2.A we state a result of [Xu and Yang, 2018] leveraging the strong convexity of the set $\mathcal{C}$ with error bound type assumptions. Indeed these are known cases where additional structure on f leads to accelerated convergence rates of the (vanilla) Frank-Wolfe algorithm (see Section 1.2). Finally, Appendix 2.B generalizes the analysis to Hölder smooth functions.

Contents

2.1 Introduction to Error Bounds

Error bounds quantify function behaviours near their minimizers. As such, they offer a comprehensive framework to capture additional structure in optimization problems. Loosely speaking then, an error bound is a kind of two-body concept involving the properties of the optimization problem and the choice of a specific quantification for the error bound itself. It can also become a three-body concept when accounting for the chosen optimization algorithm (see Section 2.2).

For instance, some works focused on characterizing large classes of functions satisfying specific types of error bounds; others on designing optimization algorithms that *accelerate* by *adapting* to the error bounds properties, which are often characterized by unknown parameters. In this chapter specifically, we design specific error bounds in the Frank-Wolfe framework and we design new Frank-Wolfe algorithms that *adapt* to these error bounds.

We now provide a partial review of error bounds in Section 2.1.1. We give pointers to the Kurdika-Łojasiewicz inequality in Section 2.1.2. In Section 2.1.3, we briefly survey works using these tools to design and analyze first-order optimization methods, and refer the reader to [Nguyen, 2017] for an in-depth discussion.

2.1.1 Error Bounds

An error bound is an inequality upper bounding the distance from an arbitrary point in a test set $\mathcal{K}$ to the level set of a function in terms of the function values. For an increasing function $\phi : \mathbb{R}^+ \to \mathbb{R}^+$ such that $\Phi(0) = 0$, $c \in \mathbb{R}$ and $K \subset \mathbb{R}^d$, an error bound takes the following form

$$\text{dist}(x, [f \leq c]) \leq \phi(f(x)), \quad \forall x \in K \ . \tag{2.1}$$

The function $\phi(\cdot)$ is known as the *residual* function and dist() denotes the Euclidean distance. Hoffman [1952] proved the first type of error bounds, quantifying the distance of a point to the set of solutions to an ensemble of linear equations, then refined in [Robinson, 1975, Mangasarian, 1985, Auslender and Crouzeix, 1988] and many others.

Denote X^* the set of minimizers of f – in this dissertation we will essentially consider strictly convex functions, so that the set X^* is a singleton that we write $\{x^*\}$. When the residual ϕ is a power function and the level set of interest is the set of minimizers X^*, (2.1) are named Hölderian error bounds (HEB). For a $\mu > 0$ and $\theta \in [0, 1]$, these take the following form

$$\text{dist}(x, X^*) \leq \mu(f(x) - f(x^*))^\theta, \quad \forall x \in K \ , \tag{2.2}$$

where $\text{dist}(x, X^*) = \min_{x \in X^*} ||x - x^*||_2$. This inequality more closely describes the behaviour of a function around its minimizers. As such, error bounds play an essential role in understanding and designing optimization algorithms.

The early works of Łojasiewicz are fundamental in this vein. He first showed that inequalities (2.2) held generically for large classes of functions, *i.e.* real analytic or subanalytic functions in [Łojasiewicz, 1958, Lojasiewicz, 1965, Łojasiewicz, 1963]. After his work, Hölderian error bounds (2.2) are also named as Łojasiewicz error bounds.

From a geometrical point of view, these inequalities characterize the behavior of functions around their extrema and are then known under different variants and names, like *sharpness inequalities* [Burke and Ferris, 1993, Burke and Deng, 2002] or *strict minimum conditions*. See also the Polyak-Łojasiewciz condition [Polyak, 1963, Karimi et al., 2016b].

We now recall the Kurdyka-Łojasiewicz gradient inequality, a.k.a. the gradient-dominated inequality, which is a closely related to the error bounds (2.2) [Bolte et al., 2010, 2017, Azé and Corvellec, 2017] and has emerged as an important tool for first-order algorithms.

2.1.2 Kurdyka-Łojasiewicz Inequality

Definition 2.1.1 (Łojasiewicz Gradient Inequality)**.** *Consider a differentiable function f, a critical point x^* and a neighborhood $\mathcal{K} \subset \mathbb{R}^d$. f satisfies a Łojasiewicz gradient inequality in the neighborhood $\mathcal{K}$ or x^* if there exists $c > 0$ and $\theta \in [1/2, 1[$ such that*

$$|f(x) - f(x^*)|^\theta \leq c||\nabla f(x)|| \quad \forall x \in \mathcal{K}. \tag{2.3}$$

Łojasiewicz first showed inequalities of this type for real analytic and subanalytic functions. See [Lojasiewicz, 1965, §18, Proposition 1] and [Bierstone and Milman, 1988, Proposition 6.8.] which states the Łojasiewicz gradient inequality with $\theta \in]0, 1[$. [Kurdyka, 1998, §2 Theorem ŁI] extended Łojasiewicez Gradient Inequality to $\mathcal{C}^1$ functions whose graph belong to an o-minimal structure. Crucially then Bolte et al. [2007] extended (2.3) to some class of non-differential functions, replacing $||\nabla f(x)||$ in (2.3) by the *non-smooth slope* [Bolte et al., 2007, (4)] $||\partial f(x)|| = \inf\{||d|| \ : \ d \in \partial f(x)\}$. In particular [Bolte et al., 2007, Theorem 3.1] extend (2.3) to continuous subanalytical functions and [Bolte et al., 2007, Theorem 3.3] to the class of convex lower semi-continuous subanalytic functions.

For convex functions, the Łojasiewicz inequality can be understood as a local generalization of strong convexity in the sense that the strong-convexity quadratic lower-bound on f at x^* implies that (2.3) holds on $\mathcal{C}$ with $\theta = 1/2$. Indeed for a convex function and any $x \in \mathcal{C}$

$$||x - x^*|| \cdot ||\nabla f(x)|| \geq (x - x^*) \cdot \nabla f(x) \geq f(x) - f(x^*),$$

and by strong convexity of f at x^*,

$$f(x) - f(x^*) \geq \underbrace{(x - x^*) \cdot \nabla f(x^*)}_{\geq 0} + \frac{\mu}{2}||x - x^*||^2 \geq \frac{\mu}{2}||x - x^*||^2.$$

Finally for any $x \in \mathcal{C}$, combining the two we have

$$||\nabla f(x)|| \geq \sqrt{\frac{\mu}{2}}\sqrt{f(x) - f(x^*)} \ .$$

Note also that the convexity of f alone, implies that (2.3) is satisfied with $\theta = 1$, the *weak* case, with

$$f(x) - f(x^*) \leq ||\nabla f(x)||\mathcal{D}, \quad \forall x \in \mathcal{C} \ ,$$

where $\mathcal{D}$ is the diameter of $\mathcal{C}$. In particular this hints that (2.3) *at least* (because it does much more than that) continuously captures behaviors in between the structure of differentiable convex function and that of a differentiable strongly convex function.

As explained in [Bolte et al., 2007], with regularity alone (2.3) may fail or holds only in the weak sense with $\theta = 1$. Bolte et al. [2007] notably provide two examples of C^∞ functions. The other way around, structure alone may not be sufficient for (2.3) to hold, as the results of [Bolte et al., 2007] seem to require at least lower semi-continuity. (2.3) is an intricate relation

between *structure* and *regularity*.

The parameters (c, θ) are generally unknowns and hard to get. To circumvent the issue, one can either design *adaptive* methods, which leverage on the additional structure of the function given in (2.3) without knowing the specific parameters, see Section 2.1.3. Another line of work is to find KL exponent (*i.e.* the value of θ in (2.3)) for various classes of functions [Luo and Sturm, 2000, Li, 2013, Vui, 2013] and explore how mathematical operations preserve KL exponents, in other words defining a *calculus* for KL exponents. For instance, Li and Pong [2018] deduce the KL exponent of a minimum over a finite number of KL functions or Yu et al. [2019] study the effect of inf-projection. Another direction is to relate Kurdyka-Łojasiewicz with others type of error bounds for which explicit quantitative statements may be easier to get. For instance, Li and Pong [2018] notably shows that Luo-Tseng error bounds plus some mild assumption on the separation of stationary values give KL exponents of $\frac{1}{2}$.

2.1.3 Error Bounds in Optimization

Kurdyka-Łojasiewicz inequality (2.3) is a local condition that generically holds and generalizes classical structural assumptions such as strong convexity. Hence, it is a key tool for the analysis of optimization methods. Bolte et al. [2017] notably shows the use Kurduka-Łojasiewicz inequality for analyzing a variety of optimization algorithms. Some works also considered non-convex settings [Attouch and Bolte, 2009, Attouch et al., 2010, 2013, Bolte et al., 2014]. Error bounds have been used for composite problems and for alternating or splitting methods [Attouch et al., 2010, 2013, Bolte et al., 2014, Frankel et al., 2015, Karimi et al., 2016b, Zhou and So, 2017].

Roulet and d'Aspremont [2017] importantly shows that sharpness can adaptively result in accelerated convergence rates for restarts schemes of smooth gradient methods. Restart was previously shown to be heuristically efficient [Giselsson and Boyd, 2014b, O'donoghue and Candes, 2015, Su et al., 2016] but without improved computational guarantees. Other works considered sharpness for restart schemes but dit no study the cost of adaptation [Nemirovskii and Nesterov, 1985a] or were not adaptive [Liu and Yang, 2017] to the error bounds parameters. This motivated our work on *restarting* Frank-Wolfe algorithms.

In the context of Frank-Wolfe algorithms, [Beck and Shtern, 2017] show that, for polytopes, when replacing the strong-convexity assumption by a quadratic errror bound, *i.e.* Hölderian error bound with $r = 2$, the away-step Frank-Wolfe enjoys linear convergence rates. Our work below considers any type of Hölderian behavior, not just a localization of strong convexity. In particular, we specifically design error bounds where the residual function is replaced by Wolfe gaps. We propose a restart scheme argument that captures the same improved convergence rates as in [Roulet and d'Aspremont, 2017, Roulet, 2017] but in the setting of Frank-Wolfe algorithms. A very important consequence of this chapter is that these acceleration results Frank-Wolfe algorithms are *adaptive* to error bound parameters.

2.2 Hölderian Error Bounds for Frank-Wolfe

Recall that we consider the following optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathcal{C} \end{array} \tag{2.4}$$

in the variables $x \in \mathbb{R}^n$, where $\mathcal{C} \subset \mathbb{R}^n$ is a compact convex set and $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function. Let X^* be the set of minimizers of f over $\mathcal{C}$ and we will consider strictly convex convex functions so that $X^* = \{x^*\}$. We assume that the following linear minimization oracle

$$\mathrm{LP}_{\mathcal{C}}(x) \triangleq \operatorname*{argmin}_{z \in \mathcal{C}} x^T z \tag{2.5}$$

can be computed efficiently.

2.2.1 Wolfe Gaps

By assumption here, we have $\mathcal{C} = \mathbf{Co}(\mathbf{Ext}(\mathcal{C}))$ where $\mathbf{Co}(\cdot)$ is the convex hull, $\mathbf{Ext}(\cdot)$ the set of extreme points, and Carathéodory's theorem shows that every point x of $\mathcal{C}$ can be written as a convex combination of at most $n+1$ points in $\mathbf{Ext}(\mathcal{C})$ although a given representation can contain more points. We call these points the *support of* x in $\mathcal{C}$. We say that a support $\mathcal{S}$ is *proper* when the weights that compose the convex combination of x are all positive. We now define the *strong-Wolfe gap* as follows.

Definition 2.2.1 (Strong-Wolfe Gap)**.** *Let f be a smooth convex function, $\mathcal{C}$ a polytope, and let $x \in \mathcal{C}$ be arbitrary. Then the* strong-Wolfe gap $w(x)$ over $\mathcal{C}$ *is defined as*

$$w(x) \triangleq \min_{S \in \mathcal{S}_x} \max_{y \in S, z \in \mathcal{C}} \nabla f(x)^T (y - z), \tag{2.6}$$

where $x \in \mathbf{Co}(S)$ and $S_x = \{S \mid S \subset \mathbf{Ext}(\mathcal{C}),$ is finite and x a proper combination of the elements of $S\}$, the set of proper supports of x. We also write

$$w(x, S) \triangleq \max_{y \in S, z \in \mathcal{C}} \nabla f(x)^T (y - z)$$

given $S \in \mathcal{S}_x$.

By construction, we have $w(x) \leq w(x, S)$. Note also that for $x \in \mathcal{C}$, the quantity $w(x, S)$ is the sum of the Frank-Wolfe dual gap with the away dual gap in [Lacoste-Julien and Jaggi, 2015a] as shows the following decomposition

$$w(x, S) = \underbrace{\max_{y \in S} \nabla f(x)^T (y - x)}_{\text{away or Wolfe (dual) gap}} + \underbrace{\max_{z \in \mathcal{C}} \nabla f(x)^T (x - z)}_{\text{Frank-Wolfe (dual) gap}}. \tag{2.7}$$

Note that only $w(x, S)$ is observed in practice, but we use $w(x)$ to simplify the primal bounds and the convergence proof. Also we write the Frank-Wolfe (dual) gap as

$$g(x) = \max_{z \in \mathcal{C}} \nabla f(x)^T (x - z). \tag{2.8}$$

We first show the following lemma on $w(x, S)$ and $w(x)$.

Lemma 2.2.2. *Let $x \in \mathcal{C}$. A finite set $S = \{v_i \mid i \in I\}$ with $v_i \in \mathbf{Ext}(\mathcal{C})$ for some finite index set I, is a proper support of x if*

$$x = \sum_{i \in S} \lambda_i v_i, \quad \text{where } \mathbf{1}^T \lambda = 1 \text{ and } \lambda_i > 0 \text{ for all } i \in I.$$

For such a proper support S of x, we have that $w(x, S) = 0$ if and only if x is an optimal solution of problem (2.4). In particular, $w(x) = 0$ if and only if x is an optimal solution of problem (2.4).

Proof. We can split $w(x, S)$ in two parts, with

$$w(x, S) = \max_{y \in S} \nabla f(x)^T (y - x) + \max_{z \in \mathcal{C}} \nabla f(x)^T (x - z)$$

It is easy to see that both summands are nonnegative if $x \in C$. Here $g(x) \triangleq \max_{z \in \mathcal{C}} \nabla f(x)(x - z)$ is the usual Wolfe gap. When x is an optimal solution of problem (2.4), first order optimality conditions implies that $\nabla f(x)^T (x - v) \leq 0$ for all $v \in \mathcal{C}$. Since this last quantity is exactly zero when $v = x$, we have $g(x) = 0$.

On the other hand let $h(x) \triangleq \max_{y \in S} \nabla f(x)^T (y - x)$, and suppose x is optimal. If $\nabla f(x) = 0$ we immediately get $h(x) = 0$. Suppose then $\nabla f(x) \neq 0$, since x is optimal, $\nabla f(x)^T (x - v_i) \leq 0$ for all v_i and we can write

$$\begin{aligned} x &= \sum_{\{i : \nabla f(x)^T (x - v_i) = 0\}} \lambda_i v_i + \sum_{\{i : \nabla f(x)^T (x - v_i) < 0\}} \lambda_i v_i \\ &= (1 - \mu) z_1 + \mu z_2 \end{aligned}$$

for some $0 \leq \mu \leq 1$, where $\nabla f(x)^T (x - z_1) = 0$ and $\nabla f(x)^T (x - z_2) < 0$. Now $0 = \nabla f(x)^T (x - x) = \mu \nabla f(x)^T (x - z_2)$ implies $\mu = 0$, hence $\nabla f(x)^T (x - v_i) = 0$ for all $i \in S$, so $h(x) = 0$. Thus we obtain, x optimal implies $w(x) = 0$. Conversely, we have

$$\begin{aligned} f(x) - f^\star &\leq \nabla f(x)^T (x - x^\star) \\ &\leq \max_{z \in \mathcal{C}} \nabla f(x)^T (x - z) \\ &\leq \max_{y \in S, z \in \mathcal{C}} \nabla f(x)^T (y - z) \\ &= w(x, S) \end{aligned}$$

by convexity (where $x^\star$ is any optimal solution), and the fact that $x \in \mathbf{Co}(S)$. Hence $w(x, S) = 0$ implies x optimal. The corollary on $w(x)$ immediately follows by construction. ∎

We will use this notion of curvature for analyzing those algorithms utilizing away steps (Algorithm 5). Note that C_f^A implicitly considers f to be defined on the Minkowski sum $\mathcal{C}^A \triangleq \mathcal{C} + (\mathcal{C} - \mathcal{C})$. Similarly (standard) *curvature* C_f [Lacoste-Julien and Jaggi, 2015a, Appendix C] is defined as

$$C_f \triangleq \sup_{\substack{x, v \in \mathcal{C} \\ \eta \in [0,1] \\ y = x + \eta(v - x)}} \frac{2}{\eta^2} (f(y) - f(x) - \eta \langle \nabla f(x), v - x \rangle), \tag{2.9}$$

and is used to bound the complexity of the classical Frank-Wolfe method (Algorithms 9 and 7).

2.2.2 Wolfe Error Bounds

We now introduce growth conditions used to bound the complexity of our variant of the Frank-Wolfe algorithm when solving the constrained optimization problem in (2). Let $\mathcal{C}$ be a general compact convex set, the following condition will be at the core of our complexity analysis. Note that similarly as the Kurdyka-Łojasiewciz inequality (as opposed to the Hölderian error bounds), the strong-Wolfe gap is formulated as an upper-bound on the primal gap $f(x) - f(x^*)$.

Definition 2.2.3 (Strong-Wolfe primal bound)**.** *Let K be a compact neighborhood of X^* in $\mathcal{C}$, where X^* is the set of solutions of the constrained optimization problem (2). A function f satisfies a r-strong-Wolfe primal bound on K, if and only if there exists $r \geq 1$ and $\mu > 0$ such that for all $x \in K$*

$$f(x) - f^* \leq \mu w(x)^r, \tag{2.10}$$

and f^* its optimal value.

In the next section, provided f is a smooth convex function, we will show for instance that $r = 2$ above guarantees linear convergence of our variant of the away-steps Frank-Wolfe algorithm. This 2-strong-Wolfe primal bound holds for example when f is strongly convex over a polytope, which corresponds to the linear convergence bound in [Lacoste-Julien and Jaggi, 2015a], hence the following observation.

Observation 2.2.4 (f strongly convex and $\mathcal{C}$ a polytope)**.** *The results in [Lacoste-Julien and Jaggi, 2015a, Theorem 6 in Eq (28)] show that when f is strongly convex and $\mathcal{C}$ is a polytope then there exists $\mu_f^A > 0$ such that for all $x \in \mathcal{C}$*

$$f(x) - f^* \leq \frac{w(x)^2}{2\mu_f^A},$$

hence condition (2.2.3) holds with $r = 2$ in this case.

The fact that $w(x) = 0$ if and only if $f(x) = f^*$ means that, in principle, the Łojasiewicz factorization lemma [Bolte et al., 2007, §3.2.] could be used to show that condition (2.10) holds generically but with unobservable parameters. These parameters are inherently hard to infer because (2.10) combines the properties of f and $\mathcal{C}$, not distinguishing between the contribution of the function from that of the structure of the constrained set (a polytope for instance).

Hence, although (2.10) has an appealing succinct form, our results will rely on the combination of a more classical Hölderian error bound (in Definition 2.2.7) defined on f, and a *scaling inequality* (defined below in Definition 2.2.5), essentially driven by the structure of the set $\mathcal{C}$. The combination of these two inequalities leads to a r-strong-Wolfe primal bound. We first state the *scaling inequality* relative to the strong-Wolfe gap $w(x)$ that we will use in the context of the the away step variant of the Frank-Wolfe algorithm.

Definition 2.2.5 (δ-scaling)**.** *A convex set $\mathcal{C}$ satisfies a* scaling inequality *if there exists $\delta(\mathcal{C}) > 0$ such that for all $x \in \mathcal{C} \setminus X^*$ and all differentiable convex function f,*

$$w(x) \geq \delta(\mathcal{C}) \max_{x^* \in X^*} \left\langle \nabla f(x); \frac{x - x^*}{||x - x^*||} \right\rangle. \tag{Scaling}$$

Here again, the strong-Wolfe gap $w(x)$ is the minimum over all proper supports of x of the scalar product of the (negative) gradient with the pairwise direction formed by the difference of the Frank-Wolfe vertex and the away vertex. Hence the δ-scaling inequality compares the worst pairwise FW direction with the normalization of the direction $x^* - x$. Notably this condition is known to hold when $\mathcal{C}$ is a polytope, with Lacoste-Julien and Jaggi [2015a] showing the following result (see also [Gutman and Pena, 2018, Pena and Rodriguez, 2018] for a simpler variant).

Lemma 2.2.6 ([Lacoste-Julien and Jaggi, 2015a])**.** *A polytope satisfies the δ-scaling inequality with $\delta(\mathcal{C}) = PWidth(\mathcal{C})$, where $PWidth(\mathcal{C})$ is the pyramidal width.*

We now recall the definition of the Hölderian error bound for a function f on problem (2) [Hoffman, 1952, Lojasiewicz, 1965, Łojasiewicz, 1993, Bolte et al., 2007] (see e.g., [Roulet and d'Aspremont, 2017] for more detailed references).

Definition 2.2.7 (Hölderian error bound (HEB))**.** *Consider a convex function f and K a compact neighborhood of X^* in $\mathcal{C}$. For optimization problem (2), f satisfies a (θ, c)-HEB on K if there exists $\theta \in [0, 1]$ and $c > 0$ such that for all $x \in K$*

$$\min_{x^* \in X^*} ||x - x^*|| \leq c(f(x) - f^*)^\theta. \tag{HEB}$$

The Hölderian error bound (HEB) locally quantifies the behavior of f around the constrained optimum of problem (2.4). A similar condition was used to show improved convergence rates for unconstrained optimization in e.g., [Nemirovskii and Nesterov, 1985c, Attouch et al., 2014, Frankel et al., 2015, Karimi et al., 2016b, Bolte et al., 2017, Roulet and d'Aspremont, 2017, Li and Pong, 2018]. Note, as we reviewed in Section 2.1, that strong convexity implies θ-HEB with $\theta = 1/2$ so (HEB) can be seen as a generalization of strong convexity. Here θ will allow us to interpolate between sub-linear and linear convergence rates.

Finally, we show that when Problem (2) satisfies both δ-Scaling and (θ, c)-HEB, the $(1 - \theta)^{-1}$-Strong-Wolfe primal bound in (2.10) holds.

Lemma 2.2.8. *Assume f is a differentiable convex function satisfying (θ, c)-HEB on K, and that $\mathcal{C}$ satisfies δ-Scaling inequality. Then for all $x \in K$*

$$f(x) - f^* \leq \left(\frac{c}{\delta}\right)^r w(x)^r,$$

with $r = \frac{1}{1-\theta}$ and f^* the objective value at constrained optima.

Proof. Assume we have (θ, c)-HEB on K. For $x \in K \backslash X^*$, by convexity, with $\tilde{x} \in \operatorname{argmin}_{x^* \in X^*} ||x - x^*||$

$$f(x) - f^* \leq \frac{\langle \nabla f(x); x - \tilde{x} \rangle}{||x - \tilde{x}||} ||x - \tilde{x}||.$$

Hence applying (θ, c)-HEB leads to

$$\begin{aligned} f(x) - f^* &\leq c \frac{\langle \nabla f(x); x - \tilde{x} \rangle}{||x - \tilde{x}||} \Big(f(x) - f^*\Big)^\theta \\ &\leq c \max_{x^* \in X^*} \frac{\langle \nabla f(x); x - x^* \rangle}{||x - x^*||} \Big(f(x) - f^*\Big)^\theta, \end{aligned} \tag{2.11}$$

from which we obtain

$$f(x) - f^* \leq c^{\frac{1}{1-\theta}} \max_{x^* \in X^*} \left(\frac{\langle \nabla f(x); x - x^* \rangle}{||x - x^*||}\right)^{\frac{1}{1-\theta}}.$$

Combining this with the δ-scaling inequality, we have

$$f(x) - f^* \leq \left(\frac{c}{\delta}\right)^{\frac{1}{1-\theta}} w(x)^{\frac{1}{1-\theta}},$$

and the desired result. ∎

In the next section, varying values of $r \in [1, 2]$ in (2.10) allow to produce sub-linear complexity bounds of the form $\mathcal{O}(1/\epsilon^{1/(2-r)})$, continuously interpolating between the known sub-linear $\mathcal{O}(1/\epsilon)$ and a linear convergence rate. For simplicity of exposition, we will always pick $K = \mathcal{C}$ in what follows. We also write $\mathbf{Int}(\cdot)$ for the interior of a set and $\mathbf{ReInt}(\cdot)$ for its relative interior.

2.2.3 Discussion

As it will be developed in the following sections, when strong-Wolfe error bounds hold – under a *non-weak* form, *i.e.* with $r > 1$ – we can accelerate our version of the Frank-Wolfe algorithms. This suggests a new perspective for determining which structure in the function f or the constraint set $\mathcal{C}$ might lead to the acceleration of Frank-Wolfe algorithms.

However, so far, we derived strong-Wolfe error bound by combining a classical error bound condition (*i.e.* involving only functional structure) with a scaling inequality which is available for polytopes only, and actually stems from the specific analysis of the Frank-Wolfe algorithms on such constraint sets. It hence remains to explore arguments to derive Wolfe error bounds directly from sub-analytical arguments in the same vein as the Kurdyka-Łojasiwicz inequality.

In particular, no acceleration result (w.r.t. the general $\mathcal{O}(1/T)$ for compact convex sets) is known for the Frank-Wolfe algorithms when the constraint set $\mathcal{C}$ is not uniformly convex (see our results in Chapter 3) or a polytope.

Hence many highly structured constraint sets are not known to provide accelerated theoretical guarantees besides the convergence rate of $\mathcal{O}(1/T)$ that holds for any compact convex set. For instance, there is no enhanced asymptotic convergence rates for the intersection of a ℓ_2 ball with a ℓ_1 ball nor for group-lasso balls.

Note that the strong-Wolfe gap $w(x)$ combines the algorithm specificity (*i.e.* the pair-wise direction), the constraint set $\mathcal{C}$ and the function f. As such, it is a structured object that is hard to analyse with sub-analytical notions. However, for the very same reason (*i.e.* the structure), it may provide a good perspective for understanding which algorithmic versions of the Frank-Wolfe algorithm may accelerate on specific constraint sets structures.

With this type of connection, Beck and Shtern [2017] provide a convergence conditioning of Away-steps Frank-Wolfe algorithms (under strong-convexity like assumption) that differs from the geometrical Pyramidal Width. Here we hope that Wolfe error bounds will give new insights in the interplay between the structure of the constraint set and acceleration of the Frank-Wolfe algorithms.

2.3 Fractional Away-Step Frank-Wolfe Algorithm

Here, we present a new variant of the Conditional Gradients method using the scaling argument of the parameter-free Lazy Frank-Wolfe variant in [Braun et al., 2017a, 2018], together with a restart scheme similar to that used for gradient methods in e.g., [Nemirovskii and Nesterov, 1985b, Giselsson and Boyd, 2014a, O'donoghue and Candes, 2015, Fercoq and Qu, 2016, Roulet and d'Aspremont, 2017]. This yields an algorithm that dynamically adapts to the local properties of the function and the feasible region around the optimum. The convergence proof relies on two key conditions. One is a scaling inequality (Definition 2.2.5) used to characterize the regularity of $\mathcal{C}$ in many Frank-Wolfe complexity bounds which holds on e.g., polytopes and strongly convex sets. The other is a local growth condition which is shown to hold generically for sub-analytic functions by the Łojasiewicz factorization lemma (see e.g., [Bolte et al., 2007]) and controls for example the impact of restart schemes as in [Roulet and d'Aspremont, 2017].

Earlier work showed that a sharpness condition derived from the Łojasiewicz lemma could be used to improve convergence rates of gradient methods (see e.g., [Nemirovskii and Nesterov, 1985a, Bolte et al., 2007, Karimi et al., 2016a] for an overview), however these methods required exact knowledge of the constants appearing in the condition to achieve improved rates. In

practice however, these constants are typically not observed. In contrast to this, as in [Roulet and d'Aspremont, 2017, Chen et al., 2018], we show using robust restart schemes that our algorithm does not require knowledge of these constants, thus making it essentially parameter-free.

We focus on the case where $\mathcal{C}$ is a polytope and f a smooth convex function. This means in particular that condition (Scaling) holds. We now state the Fractional Away-Step Frank-Wolfe method as Algorithm 5, a variant of the Away-Step Frank-Wolfe algorithm, tailored for restarting. It can be seen as the inner loop of [Braun et al., 2018, Algorithm 1], which together with a restart scheme leads to a simple version of [Braun et al., 2018, Algorithm 1] (without the cheaper Weak Separation Oracle that replaces Linear Minimization Oracle).

Algorithm 5 Fractional Away-Step Frank-Wolfe Algorithm

Input: A smooth convex function f with curvature C_f^A. Starting point $x_0 = \sum_{v \in \mathcal{S}_0} \alpha_0^v v \in \mathcal{C}$ with support $\mathcal{S}_0 \subset \mathbf{Ext}(\mathcal{C})$. LP oracle (2.5) and schedule parameter $\gamma > 0$.

$t := 0$
while $w(x_t, \mathcal{S}_t) > e^{-\gamma} w(x_0, \mathcal{S}_0)$ **do**
  $v_t := \mathrm{LP}_{\mathcal{C}}(\nabla f(x_t))$ and $d_t^{FW} := v_t - x_t$
  $s_t := \mathrm{LP}_{S_t}(-\nabla f(x_t))$ with S_t current active set and $d_t^{Away} := x_t - s_t$
  if $-\nabla f(x_t)^T d_t^{FW} > e^{-\gamma} w(x_0, \mathcal{S}_0)/2$ **then**
    $d_t := d_t^{FW}$ with $\eta_{\max} := 1$
  else
    $d_t := d_t^{Away}$ with $\eta_{\max} := \frac{\alpha_t^{s_t}}{1-\alpha_t^{s_t}}$
  end if
  $x_{t+1} := x_t + \eta_t d_t$ with $\eta_t \in [0, \eta_{\max}]$ via line-search
  Update active set $\mathcal{S}_{t+1}$ and coefficients $\{\alpha_{t+1}^v\}_{v \in \mathcal{S}_{t+1}}$
  $t := t + 1$
end while

Output: $x_t \in \mathcal{C}$ such that $w(x_t, \mathcal{S}_t) \leq e^{-\gamma} w(x_0, \mathcal{S}_0)$

In the following we will call a step a *full-progress step* if it is a Frank-Wolfe step or an Away step that is not a drop step, i.e., when $\eta_t < \alpha_{s_t}/(1 - \alpha_{s_t})$. The support $\mathcal{S}_t$ and the weights α_t are updated exactly as in [Lacoste-Julien and Jaggi, 2015a, Away-Steps Frank-Wolfe]. Algorithm 5 depends on a parameter $\gamma > 0$ which explicitly controls the number of iterations needed for the algorithm to stop. In particular, a large value of γ will increase the number of iterations and when γ converges to infinity, Algorithm 5 tends to behave exactly like the classical Frank-Wolfe, (i.e., it never chooses the away direction as an update direction, see Appendix 2.C for a proof).

Proposition 2.3.1 below gives an upper bound on the number of iterations required for Algorithm 5 to reach a given target gap $w(x_T, \mathcal{S}_T) \leq w(x_0, \mathcal{S}_0)e^{-\gamma}$. The assumption $e^{-\gamma} w(x_0, \mathcal{S}_0)/2 \leq C_f^A$ in this proposition measures the complexity of a burn-in phase whose cost is marginal as shown in Proposition 2.3.2.

Proposition 2.3.1 (Fractional Away-Step Frank-Wolfe Complexity)**.** *Let f be a smooth convex function with away curvature C_f^A such that the r-strong-Wolfe primal bound in (2.10) holds on $\mathcal{C}$ (with $1 \leq r \leq 2$ and $\mu > 0$). Let $\gamma > 0$ and assume $x_0 \in \mathcal{C}$ is such that $e^{-\gamma} w(x_0)/2 \leq C_f^A$.*

Algorithm 5 outputs an iterate $x_T \in \mathcal{C}$ such that

$$w(x_T, \mathcal{S}_T) \leq w(x_0, \mathcal{S}_0) e^{-\gamma}$$

after at most

$$T \leq |\mathcal{S}_0| - |\mathcal{S}_T| + 16 e^{2\gamma} C_f^A \mu w(x_0, \mathcal{S}_0)^{r-2}$$

iterations, where $\mathcal{S}_0$ and $\mathcal{S}_T$ are the supports of respectively x_0 and x_T.

Proof. Because of the test criterion in line 5, the update direction d_t satisfies (writing $r_t \triangleq -\nabla f(x_t)$),

$$r_t^T d_t > e^{-\gamma} w(x_0, \mathcal{S}_0)/2 \ .$$

Indeed, this holds by definition when choosing the FW direction, otherwise (2.7) yields

$$w(x_t, \mathcal{S}_t) = r_t^T d_t^{FW} + r_t^T d_t^{Away} > e^{-\gamma} w_0,$$

(writing $w_0 \triangleq w(x_0, \mathcal{S}_0)$ to simplify notations) so that

$$r_t^T d_t^{Away} > e^{-\gamma} w_0 - r_t^T d_t^{FW} \geq e^{-\gamma} w_0 - e^{-\gamma} w_0/2 = e^{-\gamma} w_0/2.$$

Using curvature in (1.16), we have for d_t,

$$f(x_t + \eta d_t) \leq f(x_t) + \eta \nabla f(x_t)^T d_t + \frac{\eta^2}{2} C_f^A \ ,$$

which implies

$$f(x_t) - f(x_t + \eta d_t) \geq \eta r_t^T d_t - \frac{\eta^2}{2} C_f^A \ .$$

We can lower bound progress $f(x_t) - f(x_{t+1})$ with $x_{t+1} = x_t + \eta d_t$ at each iteration for full-progress steps. For Frank-Wolfe steps,

$$\begin{aligned} f(x_t) - f(x_{t+1}) &\geq \max_{\eta \in [0,1]} \left\{ \eta r_t^T d_t - \frac{\eta^2}{2} C_f^A \right\} \\ &\geq \max_{\eta \in [0,1]} \left\{ \eta e^{-\gamma} w_0/2 - \frac{\eta^2}{2} C_f^A \right\} \end{aligned}$$

Hence because of exact line-search (in practice many alternatives exist which will not affect the convergence proofs, see e.g., [Pedregosa et al., 2018]), assuming $e^{-\gamma} w_0/2 \leq C_f^A$ holds,

$$f(x_t) - f(x_{t+1}) \geq \frac{w_0^2}{8 C_f^A e^{2\gamma}}. \tag{2.12}$$

For all away steps, we have

$$f(x_t) - f(x_t + \eta d_t) \geq \max_{\eta \in [0, \eta_{\max}]} \left\{ \eta e^{-\gamma} w_0/2 - \frac{\eta^2}{2} C_f^A \right\}.$$

Yet for away steps that are not drop steps, assuming $e^{-\gamma} w_0/2 \leq C_f^A$ again the optimum is obtained for $0 < \eta^* < \eta_{\max}$, and the same conclusion as in (2.12) for Frank-Wolfe steps follows.

Write $T = T_d + T_f$ the number of iterations for Algorithm 5 to finish. Here T_d denotes the number of drop steps, while T_f stands for the number of full-progress steps. Hence we have,

$$\begin{aligned} f(x_0) - f(x_T) &= \sum_{t=0}^{T-1} f(x_t) - f(x_{t+1}) \\ &\geq T_f \frac{w_0^2}{8C_f^A e^{2\gamma}}. \end{aligned}$$

Because f satisfies a r-Strong-Wolfe primal gap on $\mathcal{C}$ we have when $x_0 \in \mathcal{C}$,

$$f(x_0) - f(x_T) \leq f(x_0) - f^* \leq \mu w(x_0)^r \leq \mu w(x_0, \mathcal{S}_0)^r,$$

by definition of $w(x)$. We then get an upper bound on the number T_f of full-progress steps

$$T_f \leq 8C_f^A e^{2\gamma} \mu w_0^{r-2} \ .$$

Finally writing $|\mathcal{S}_0|$ (resp. $|\mathcal{S}_T|$) the size of the support of x_0 (resp. x_T), and T_{FW} the number of Frank-Wolfe steps which add a new vertex to an iterate of the Fractional-Away-Step Frank-Wolfe Algorithm, we get $T_{FW} \leq T_f$ and the size of the support $\mathcal{S}_t$ of x_t satisfies $|\mathcal{S}_0| - T_d + T_{FW} = |\mathcal{S}_T|$ hence

$$|\mathcal{S}_0| - |\mathcal{S}_T| + T_f \geq T_d,$$

and we finally get $T \leq |\mathcal{S}_0| - |\mathcal{S}_T| + 16C_f^A e^{2\gamma} \mu w_0^{r-2}$. ■

The following observation shows that the assumption $e^{-\gamma} w(x_0, \mathcal{S}_0)/2 \leq C_f^A$ in Proposition 2.3.1 has a marginal impact on complexity.

Proposition 2.3.2 (Burn-in phase)**.** *After at most*

$$8\frac{e^\gamma}{\gamma} \ln \frac{w(x_0, \mathcal{S}_0)}{2C_f^A} + |\mathcal{S}_0|,$$

cumulative iterations of Algorithm 5, with constant schedule parameter $\gamma > 0$, we obtain a point x such that $e^{-\gamma} w(x, \mathcal{S})/2 \leq C_f^A$.

Proof. The proof closely follows that of Proposition 2.3.1. Let $w_0 = w(x_0, \mathcal{S}_0)$ and suppose that $e^{-\gamma} w_0/2 > C_f^A$. Then by curvature, for every full progress step, we would have an optimal step length $\eta_t \geq 1$, which we cap to 1 as we form convex combinations. Hence with $\eta_t = 1$ in this case we have

$$f(x_t) - f(x_{t+1}) \geq \eta_t e^{-\gamma} w_0/2 - \frac{\eta_t^2 C_f^A}{2} \geq e^{-\gamma} w_0/2 - \frac{C_f^A}{2} \geq e^{-\gamma} w_0/4.$$

Note that Lemma 2.2.2 implies that when the exit condition is not satisfied, x_t cannot be optimal so the left-hand side above cannot be zero. Moreover, via the strong-Wolfe gap we have

$$f(x_0) - f(x^*) \leq w_0 \ .$$

Writing T the number of iterations of the Algorithm 5 before it stopped, with same notation as in Proposition 2.3.1, combining the equations above yields

$$T_f e^{-\gamma} w_0/4 \leq f(x_0) - f(x_T) \leq f(x_0) - f(x^*) \leq w_0.$$

Hence

$$T_f e^{-\gamma} w_0/4 \leq w_0,$$

and $T_f \leq 4e^{\gamma}$. Also

$$T = T_d + T_f \leq 2T_f + |\mathcal{S}_0| - |\mathcal{S}_T|,$$

so that

$$T \leq 8e^{\gamma} + |\mathcal{S}_0|.$$

Because x_T is the output of Algorithm 5, we have $w(x_T, \mathcal{S}_T) < e^{-\gamma} w_0$. Write N the smallest integer such that $e^{-N\gamma} w_0 \leq 2C_f^A e^{\gamma}$ and $\hat{x}_i$ (for $0 \leq i \leq N$) the output of the i^{th} call to Algorithm 5. It is sufficient that N satisfies

$$N \geq \frac{1}{\gamma} \ln \frac{w_0}{2C_f^A} - 1.$$

Similarly write $i_0 \leq N$ the first integer such that $w(\hat{x}_{i_0}) < 2C_f^A e^{\gamma}$. If $i_0 = N$, each of the first N calls to Algorithm 5 runs in less than $8e^{\gamma} + |\mathcal{S}_{\hat{x}_i}| - |\mathcal{S}_{\hat{x}_{i+1}}|$ iterations. And we finally need at most

$$8\frac{e^{\gamma}}{\gamma} \ln \frac{w_0}{2C_f^A} + |\mathcal{S}_0| \quad \text{iterations.}$$

Otherwise $i_0 < N$ and hence $e^{-i_0\gamma} w_0 \geq C_f^A e^{\gamma}$ from which it follows that

$$i_0 \leq \frac{1}{\gamma} \ln \frac{w_0}{2C_f^A e^{\gamma}},$$

and similarly, each call before the i_0^{th} of Algorithm 5 requires also a bounded number of iterations $8e^{\gamma} + |\mathcal{S}_{\hat{x}_i}| - |\mathcal{S}_{\hat{x}_{i+1}}|$ so that we need at most

$$8\frac{e^{\gamma}}{\gamma} \ln \frac{w(x_0, \mathcal{S}_0)}{2C_f^A e^{\gamma}} + |\mathcal{S}_0| \quad \text{iterations,}$$

which is the desired result. ■

2.4 Restart Schemes

Consider a point x_{k-1} with strong-Wolfe gap $w(x_{k-1}, \mathcal{S}_{k-1})$. Algorithm 5 with parameter $\gamma_k > 0$, outputs a point x_k and we write

$$x_k \triangleq \mathcal{F}(x_{k-1}, w(x_{k-1}, \mathcal{S}_{k-1}), \gamma_k).$$

Following [Roulet and d'Aspremont, 2017] we define *scheduled restarts* for Algorithm 5 as follows.

Algorithm 6 Scheduled restarts for Fractional Away-step Frank-Wolfe

Input: $\tilde{x}_0 \in \mathbb{R}^n$ and a sequence $\gamma_k > 0$ and $\epsilon > 0$.
Burn-in phase: compute x_0 via $8\frac{e^{\gamma}}{\gamma} \ln \frac{w(x_0, \mathcal{S}_0)}{2C_f^A} + |\mathcal{S}_0|$ steps of Algorithm 5.
while $w(x_{k-1}) > \epsilon$ **do**

$$x_k = \mathcal{F}(x_{k-1}, w(x_{k-1}, \mathcal{S}_{k-1}), \gamma_k)$$

end while
Output: $\hat{x} := x_T$

Note that one overall burn-in phase is sufficient to ensure the condition $e^{-\gamma_i} w(x_{i-1}, \mathcal{S}_{i-1})/2 \leq C_f^A$ at each restart.

Algorithm 6 is similar to the restart scheme in [Roulet and d'Aspremont, 2017, Section 4] where a termination criterion is available. In this situation, [Roulet and d'Aspremont, 2017] show that the convergence rate of restarted gradient methods is robust to a suboptimal choice of restart scheme parameter γ. Here we also show that our restart scheme is adaptive to the unknown parameters in (θ, c)-HEB.

Importantly also, Algorithm 6 shares the same structure as the methods in [Lan et al., 2017, Braun et al., 2018] but these later methods do not tune the γ parameter. We will see below in Proposition 2.4.3 that tuning γ only has a marginal impact on the complexity bound. Note also that when $\theta \in [0, 1/2]$, the condition interpolates between the non-strongly convex function f and a strongly convex function scenarios. Note also that a linear function satisfies θ-HEB with $\theta = 1$ and in this case, FW converges in one iteration.

Theorem 2.4.1 (Rate for constant restart schemes)**.** *Let f be a smooth convex function with away curvature C_f^A. Assume $\mathcal{C}$ satisfies δ-Scaling and f is (θ, c)-HEB on $\mathcal{C}$. Let $\gamma > 0$ and assume $x_0 \in \mathcal{C}$ is such that $e^{-\gamma} w(x_0, \mathcal{S}_0)/2 \leq C_f^A$. With $\gamma_k = \gamma$, the output of Algorithm 6 satisfies ($r = \frac{1}{1-\theta}$)*

$$\begin{cases} f(x_T) - f^* \leq w_0 \dfrac{1}{\left(1 + \tilde{T} C_\gamma^r\right)^{\frac{1}{2-r}}} & \text{when } 1 \leq r < 2 \\ f(x_T) - f^* \leq w_0 \exp\left(-\dfrac{\gamma}{e^{2\gamma}} \dfrac{\tilde{T}}{8 C_f^A \mu}\right) & \text{when } r = 2 \ , \end{cases} \tag{2.13}$$

after T steps, with $w_0 = w(x_0, \mathcal{S}_0)$, $\tilde{T} \triangleq T - (|\mathcal{S}_0| - |\mathcal{S}_T|)$, and

$$C_\gamma^r \triangleq \frac{e^{\gamma(2-r)} - 1}{8 e^{2\gamma} C_f^A \mu w(x_0, \mathcal{S}_0)^{r-2}} \tag{2.14}$$

with $\mu = \frac{c}{\delta}$.

Proof. Denote by R the number of restarts in Algorithm 5 for T total iterations. By design

$$w(x_R, \mathcal{S}_R) \leq w_0 e^{-\gamma R}.$$

Because f is (θ, c)-HEB and $\mathcal{C}$ satisfies δ-Scaling, via Lemma 2.2.8, f satisfies the r-strong-Wolfe primal bound (2.10) with $r = \frac{1}{1-\theta}$. Using Proposition 2.3.1, the total number T of steps of Algorithms 5 is upper-bounded by

$$T \leq |\mathcal{S}_0| - |\mathcal{S}_T| + 8 C_f^A \mu e^{2\gamma} w_0^{r-2} \sum_{i=0}^{R-1} e^{-\gamma i (r-2)}.$$

Suppose $r < 2$, we have the following upper bound on T,

$$T \leq |\mathcal{S}_0| - |\mathcal{S}_T| + 8 C_f^A \mu e^{2\gamma} w_0^{r-2} \frac{e^{\gamma(2-r)R} - 1}{e^{\gamma(2-r)} - 1},$$

hence

$$e^{-\gamma R} \leq \frac{1}{\left(1+\tilde{T}C_\gamma^r\right)^{\frac{1}{2-r}}}.$$

Thus for $1 \leq r < 2$,

$$w(x_R, \mathcal{S}_R) \leq w_0 \frac{1}{\left(1+\tilde{T}C_\gamma^r\right)^{\frac{1}{2-r}}}.$$

The case $r = 2$ leads to

$$T \leq |\mathcal{S}_0| - |\mathcal{S}_T| + 8C_f^A \mu e^{2\gamma} R,$$

and hence

$$w(x_R, \mathcal{S}_R) \leq w_0 \exp\left(-\gamma \frac{\tilde{T}}{8C_f^A \mu e^{2\gamma}}\right),$$

which yields the desired result. ■

Corollary 2.4.2. *When $\mathcal{C}$ is a polytope and f a smooth convex function satisfying θ-HEB, rates in Theorem 2.4.1 hold. In particular when f is strongly convex, $\theta = \frac{1}{2}$ (and hence $r = 2$) and Algorithm 6 converges linearly. When f is simply smooth, $\theta = 0$ (and hence $r = 1$) and Algorithm 6 converges sub-linearly with a rate of $\mathcal{O}(1/t)$.*

Note also that for $r \to 2$, we recover the same complexity rates as for $r = 2$

$$\lim_{r\to 2} \frac{1}{\left(1+\tilde{T}C_\gamma^r\right)^{\frac{1}{2-r}}} = \exp\Big(-\frac{\gamma}{e^{2\gamma}}\frac{\tilde{T}}{8C_f^A\mu}\Big).$$

The complexity bounds in Theorem 2.4.1 depend on γ, which controls the convergence rate. Optimal choices of γ depend on r, a constant that we generally do not know nor observe. However, in the following we show that simply picking $\gamma = 1/2$ leads to optimal complexity bounds up to a constant factor. In fact, picking a constant gamma (independent of r) we also recover a simple version of [Braun et al., 2018, Algorithm 1] (without the cheaper Weak Separation Oracle that replaces the Linear Minimization Oracle).

Proposition 2.4.3 (Robustness in γ)**.** *Suppose f satisfies the r-strong-Wolfe primal bound (2.10) with $r > 0$. Write $\gamma^*(r)$ as the optimal choice of $\gamma > 0$ in the complexity bounds (2.13) of Theorem 2.4.1. Consider running Algorithm 6 with $\gamma = 1/2$ and the same assumptions as in Theorem 2.4.1, the output $\hat{x}$ satisfies*

$$h(\hat{x}) \leq \sqrt{\frac{e}{4(\sqrt{e}-1)}} w_0 \frac{1}{\left(1+\tilde{T}C_{\gamma^*(r)}^r\right)^{\frac{1}{2-r}}} \quad \text{when } 1 \leq r < 2,$$

where

$$C_\gamma^r = \frac{e^{\gamma(2-r)}-1}{8e^{2\gamma}C_f^A \mu w(x_0, \mathcal{S}_0)^{r-2}},$$

as in (2.14). When $r = 2$, we have $\gamma^*(r) = 1/2$.

Proof. When $1 \leq r < 2$, from Theorem 2.4.1 we have

$$f(x_T) - f^* \leq w_0 \frac{1}{\left(1 + \tilde{T} C_\gamma^r\right)^{\frac{1}{2-r}}}. \tag{2.15}$$

With definition of C_γ^r in (2.14), minimizing (2.15) is equivalent to maximizing (for $\gamma > 0$)

$$B(\gamma) = \Big(\frac{e^{\gamma(2-r)} - 1}{e^{2\gamma}}\Big).$$

Hence the optimum schedule parameter $\gamma^*(r)$ is

$$\gamma^*(r) = \frac{\ln(2) - \ln(r)}{2 - r} \quad \text{when } 1 \leq r < 2.$$

In particular $\gamma^*(r) \in]1/2; \ln(2)]$. Let's now show that the bound in (2.15) obtained with the optimal $\gamma^*(r)$ is comparable to the bound obtained with $\gamma = \frac{1}{2}$. The function

$$H(r) = \frac{\left(1 + \tilde{T} C_{\gamma^*(r)}^r\right)^{\frac{1}{2-r}}}{\left(1 + \tilde{T} C_{1/2}^r\right)^{\frac{1}{2-r}}}$$

is decreasing in r. Write $\tilde{C} \triangleq 8 C_f^A \mu w(x_0, \mathcal{S}_0)$, we have $C_{\gamma^*(1)}^1 = 1/(4\tilde{C})$ and $C_{1/2}^1 = \frac{\sqrt{e}-1}{e}/\tilde{C}$ hence

$$H(1) = \sqrt{\frac{1 + \frac{\tilde{T}}{\tilde{C}} \frac{1}{4}}{1 + \frac{\tilde{T}}{\tilde{C}} \frac{\sqrt{e}-1}{e}}} \leq \sqrt{\frac{e}{4(\sqrt{e} - 1)}}\,.$$

Hence, with $H(1) \geq H(r)$, we get for any $r \in [1, 2[$

$$\frac{1}{\left(1 + \tilde{T} C_{1/2}^r\right)^{\frac{1}{2-r}}} \leq \sqrt{\frac{e}{4(\sqrt{e} - 1)}} \frac{1}{\left(1 + \tilde{T} C_{\gamma^*(r)}^r\right)^{\frac{1}{2-r}}}.$$

When $r = 2$, the optimal choice for γ is $1/2$, maximizing the function $\gamma/e^{2\gamma}$. ∎

2.5 Fractional Frank-Wolfe Algorithm

In this section, we describe how Hölderian error bounds coupled with a restart scheme yield improved convergence bounds for the Frank-Wolfe algorithm.

In Section 2.3, relaxing the strong convexity of f using the (θ, c)-HEB assumption leads to improved sub-linear rates using a restart scheme for the Away step variant of the Frank-Wolfe algorithm, when the set of constraints $\mathcal{C}$ is a polytope. For these sets, away steps produce accelerated convergence rates that the (vanilla) Frank-Wolfe algorithm cannot achieve.

However, accelerated convergence holds for the vanilla Frank-Wolfe algorithm in other scenarios. For instance, when the solution of (2.4) is in the interior of the set and f is strongly convex, the convergence of the Frank-Wolfe algorithm is linear. In this vein, we define a fractional version of the Frank-Wolfe algorithm (Algorithm 7) and analyze its restart scheme (Algorithm 8) under the (θ, c)-HEB condition in Section 2.5.2.

Note that restart schemes of the Fractional Frank-Wolfe algorithm perform the very same iterations as the Frank-Wolfe algorithm. However, the restart scheme produces a much simpler proof of improved convergence bounds. The fractional variant is also the structural basis for recent competitive versions of the Frank-Wolfe algorithm [Braun et al., 2017a].

Another acceleration scenario for the Frank-Wolfe algorithm is when the set of constraints $\mathcal{C}$ is strongly convex. Under some restrictive assumption on f, the classical analysis [Levitin and Polyak, 1966, (5) in Theorem 6.1] exhibits a linear convergence rate. Recently [Garber and Hazan, 2015] have shown a general $\mathcal{O}(1/T^2)$ sub-linear rate when f and $\mathcal{C}$ are strongly convex. We will state new rates for the case where f satisfies (θ, c)-HEB and $\mathcal{C}$ is strongly convex, providing a more complete picture. Note that Chapter 3 provides the general extension of this notion of strong-convexity in the set.

For completeness, we would like to mention that δ-scaling for the away step Frank-Wolfe algorithm does not apply in the case where $\mathcal{C}$ is a strongly convex set. In fact, Lemma 2.2.6 does not hold anymore, and $PWidth$ can tend to zero in this case.

2.5.1 Restart Schemes for Fractional Frank-Wolfe

We now state the *fractional* version of the (vanilla) Frank-Wolfe algorithm. The Fractional Frank-Wolfe algorithm 7 is derived from Algorithm 5 by replacing $w(x_0, \mathcal{S}_0)$ with $g(x_0)$, as in (2.8) and dropping the away step update.

Algorithm 7 Fractional Frank-Wolfe Algorithm

Input: A smooth convex function f with curvature C_f. Starting point $x_0 \in \mathcal{C}$. LP oracle (2.5) and schedule parameter $\gamma > 0$.

1: $t := 0$
2: **while** $g(x_t) > e^{-\gamma} g(x_0)$ **do**
3: $\quad v_t := \mathrm{LP}_{\mathcal{C}}(\nabla f(x_t))$ and $d_t^{FW} := v_t - x_t$
4: $\quad x_{t+1} := x_t + \eta_t d_t^{FW}$ with $\eta_t \in [0, 1]$ via line-search
5: $\quad t := t + 1$
6: **end while**

Output: $x_t \in \mathcal{C}$ such that $g(x_t) \leq e^{-\gamma} g(x_0)$

A constant restart scheme using Algorithm 7 for its inner iteration, recovers the Scaling Frank-Wolfe algorithm [Braun et al., 2017a, Algorithm 7: Parameter-free Lazy Conditional Gradient] up to a slight reformulation with the additional Φ_t parameter. The two algorithms have the same restart structure, but the Scaling Frank-Wolfe algorithm additionally uses a weaker oracle (a so-called Weak Separation Oracle) than the Linear Optimization Oracle that we employ here. More precisely, the Scaling Frank-Wolfe algorithm does not necessarily require v_t to be the exact nor an approximate solution of the Linear Minimization Problem, but rather to satisfy the condition $\langle -\nabla f(x_t); v_t - x_t \rangle > \Phi_t e^{-\gamma}$. As a consequence, $g(x_t)$ is not computed and Φ_t is only an upper bound on $g(x_t)$. This explains the difference in line 8 of Algorithm 8.

2.5.2 Optimum in the Interior of the Feasible Set

We first recall that when the optimal solutions of (2) are in the relative interior of $\mathcal{C}$, a version of the (Scaling) inequality is automatically satisfied. (FW-Scaling) replaces $w(x)$ by $g(x)$ and can be interpreted as a scaling inequality tailored to the (vanilla) Frank-Wolfe algorithm. Note

Algorithm 8 Restart Fractional Frank-Wolfe Algorithm

Input: A smooth convex function f with curvature C_f. Starting point $x_0 \in \mathcal{C}$. $\epsilon > 0$, LP oracle (2.5) and schedule parameter $\gamma > 0$.

$t := 0$ and $\Phi_0 := g(x_0)$
while $g(x_t) > \epsilon$ **do**
  $v_t := \text{LP}_{\mathcal{C}}(\nabla f(x_t))$ and $d_t^{FW} := v_t - x_t$
  if $\langle -\nabla f(x_t); v_t - x_t \rangle > \Phi_t e^{-\gamma}$ **then**
    $x_{t+1} := x_t + \eta_t d_t^{FW}$ with $\eta_t \in [0, 1]$ via line-search
    $\Phi_{t+1} := \Phi_t$
  else
    $\Phi_{t+1} := g(x_t)$ (hence $\Phi_{t+1} < \Phi_t e^{-\gamma}$)
  end if
  $t := t + 1$
end while

that the δ parameter depends on the relative distance of the optimal set X^* to the boundary of $\mathcal{C}$. This property has already been extensively used in e.g., [Guélat and Marcotte, 1986, Garber and Hazan, 2013a, Garber and Meshi, 2016].

Lemma 2.5.1 (FW δ-scaling when optimum is in relative interior [Guélat and Marcotte, 1986])**.** *Assume $\mathcal{C}$ is convex and f convex differentiable. Assume $X^* \subset$ **ReInt**$(\mathcal{C})$ and choose a $z > 0$ such that $B(x^*, z) \cap$ **Aff**$(\mathcal{C}) \subset \mathcal{C}$ for all $x^* \in X^*$. Then for all $x \in \mathcal{C}$ such that $d(x, X^*) \leq \frac{z}{2}$ we have*

$$g(x) \geq \frac{z}{2} ||Proj_{\textbf{Aff}(\mathcal{C})}(\nabla f(x))||, \qquad \text{(FW-Scaling)}$$

*where **Aff**$(\mathcal{C})$ is the smallest affine set containing $\mathcal{C}$ and $g(x)$ is the Frank-Wolfe (dual) gap as defined in (2.8).*

Proof. Let $x \in B(x^*, \frac{z}{2}) \cap \mathcal{C}$. Write $d = \text{Proj}_{\textbf{Aff}(\mathcal{C})}(\nabla f(x))$. By assumption $B(x^*, z) \cap \textbf{Aff}(\mathcal{C}) \subset \mathcal{C}$, hence $x - \frac{z}{2}\frac{d}{||d||} \in \mathcal{C}$. Denote v the Frank-Wolfe vertex, we have $g(x) \triangleq \langle -\nabla f(x); v - x \rangle$. By optimality of v, we have

$$g(x) \geq \langle -\nabla f(x); x - \frac{z}{2}\frac{d}{||d||} - x \rangle = \frac{z}{2} ||\text{Proj}_{\textbf{Aff}(\mathcal{C})}(\nabla f(x))|| \ ,$$

which is the desired result. ∎

When $\mathcal{C}$ is full dimensional, its relative interior matches its interior and the projection operation is the identity. Stating the result in term of the relative interior allows to update the convex set $\mathcal{C}$. Indeed when an iterate hits a face F of $\mathcal{C}$, the future iterates might then remain in the convex F.

We now bound the convergence rate of Algorithm 8 in the following proposition.

Proposition 2.5.2 (Convergence Rate of Restart Fractional FW)**.** *Let f be a smooth convex function with curvature C_f as defined in (2.9), satisfying (θ, c)-HEB on $\mathcal{C}$. Assume there exists $z > 0$ such that $B(x^*, z) \subset \mathcal{C}$ for all $x^* \in X^*$. Let $\gamma > 0$ and assume x_0 is such that $e^{-\gamma} g(x_0) \leq C_f$ and $f(x_0) - f^* \leq (\frac{z}{2})^{\frac{1}{\theta}}$ (burn-in phase). Then the output of Algorithm 6*

satisfies ($r = \frac{1}{1-\theta}$)

$$\begin{cases} f(x_T) - f^* \leq g_0 \dfrac{1}{\left(1 + TC_\gamma^r\right)^{\frac{1}{2-r}}} & \text{when } 1 \leq r < 2 \\ f(x_T) - f^* \leq g_0 \exp\left(-\dfrac{\gamma}{e^{2\gamma}} \dfrac{T}{8C_f\mu}\right) & \text{when } r = 2 \ , \end{cases}$$

after T *steps, with* $g_0 = g(x_0)$. *Also*

$$C_\gamma^r \triangleq \frac{e^{\gamma(2-r)} - 1}{2e^{2\gamma} C_f \mu g(x_0)^{r-2}}$$

with $\mu = \frac{c}{\delta}$.

Proof. First note that for all t, we have $d(x_t, X^*) \leq \frac{z}{2}$. Indeed $f(x_t) - f^* \leq f(x_{t-1}) - f^* \leq \left(\frac{z}{2}\right)^{\frac{1}{\theta}}$. Hence by (θ, c)-HEB we have

$$\min_{x^* \in X^*} ||x_t - x^*|| \leq (f(x_t) - f^*)^\theta \leq \frac{z}{2}.$$

We can now apply lemma 2.5.1 to get for all x_t

$$g(x_t) \geq \frac{z}{2} ||\nabla f(x_t)||,$$

and as in Lemma 2.2.8, FW-Scaling and θ-HEB leads to a Wolfe primal gap (with $\mu = (cz/2)^{1/(1-\theta)} > 0$)

$$f(x) - f^* \leq \mu g(x)^r,$$

where $r = 1/(1-\theta)$. The proof then follows exactly that of Fractional Away Frank-Wolfe and its restart schemes (see Proposition 2.3.1 and Theorem 2.4.1), replacing $w(x)$ with $g(x)$. The only change comes from the upper bound on T, the number of iterations needed for Fractional Frank-Wolfe to stop. We recall the key steps to get this bound and update its value. At each iteration

$$f(x_t) - f(x_{t+1}) \geq \max_{\eta \in [0,1]} \{\eta e^{-\gamma} g(x_0) - \frac{\eta^2}{2} C_f\},$$

such that because of assumption $e^{-\gamma} g(x_0) < C_f$, we have

$$f(x_t) - f(x_{t+1}) \geq \frac{1}{2} \frac{g(x_0)^2}{e^{2\gamma} C_f}.$$

Hence on one side

$$f(x_0) - f(x_T) \geq \frac{T}{2} \frac{g(x_0)^2}{e^{2\gamma} C_f}.$$

And on the other side, using the r-Wolfe primal bound $f(x_0) - f(x_T) \leq \mu g(x_0)^r$ and finally

$$T \leq 2\mu C_f e^{2\gamma} g(x_0)^{r-2}.$$

The restart scheme is then controlled exactly as in the proof of 2.4.1. ∎

Assuming that $e^{-\gamma} g(x_0) \leq C_f$ and $f(x_0) - f^* \leq \left(\frac{z}{2}\right)^{\frac{1}{\theta}}$ simplify the statements and it is automatically satisfied after a burn-in phase. However it is fundamental to assume that there exists $z > 0$ s.t. $B(x^*, z) \subset \mathcal{C}$ for all $x^* \in X^*$. Indeed this ensures that the optimal set is in the relative interior of $\mathcal{C}$. Note also that a robustness result similar to that of Proposition 2.4.3 holds here.

Appendices

2.A Strongly Convex Constraint Set

When $\mathcal{C}$ is strongly convex, strong convexity of f leads to better convergence rate than the sub-linear $\mathcal{O}(1/T)$. The original analysis of [Levitin and Polyak, 1966, (5) in Theorem 6.1] assumes $||\nabla f(x)|| \geq \epsilon > 0$ (irrespective of the strong convexity of f) and hence (θ, c)-HEB cannot be understood as a relaxation of the assumption. This analysis provides linear convergence rate when the unconstrained minimum of f is strictly outside of $\mathcal{C}$. §2.5.2 shows linear convergence when x^* is in the interior of $\mathcal{C}$. Hence the remaining case is when the unconstrained minimum of f is in $\partial\mathcal{C}$, the boundary of $\mathcal{C}$ (an arguably rare instance).

Recently, the analysis of [Garber and Hazan, 2015] closes this gap by providing a general convergence rate of $\mathcal{O}(1/T^2)$ under a (slightly) weaker assumption than strong convexity of f [Garber and Hazan, 2015, see (2)]. The asymptotic rate regime of [Garber and Hazan, 2015] is less appealing than the linear convergence rate in [Levitin and Polyak, 1966]. However, the bound of [Garber and Hazan, 2015] benefits from much better conditioning and can easily dominate other bounds when the optimum is near $\partial\mathcal{C}$. In particular the conditioning of [Levitin and Polyak, 1966] depends on the ϵ lower bounding the norm of the gradient on the constraint set which can be arbitrarily small. The analysis of [Garber and Hazan, 2015] adapts to (θ, c)-HEB, as it was detailed in [Xu and Yang, 2018]. We recall this below for the sake of completeness.

Theorem 2.A.1. *Consider $\mathcal{C}$ an α-strongly convex set and f a convex L-smooth function. Assume (θ, c)-HEB for f. Then the iterate of the Frank-Wolfe algorithm (with exact line search or short step sizes) is such that $f(x_T) - f(x^*) = \mathcal{O}\big(1/T^{1/(1-\theta)}\big)$ for $\theta \in [0, 1[$.*

Proof. From [Garber and Hazan, 2015, Lemma 1], L-smoothness of f combines with α-strong convexity of $\mathcal{C}$ gives

$$h_{t+1} \leq h_t \cdot \max\left\{\frac{1}{2}, 1 - \frac{\alpha||\nabla f(x)||}{8L}\right\}.$$

On the other hand with (θ, c)-HEB and by convexity of f, (2.11) applies

$$\begin{aligned}\Big(f(x) - f(x^*)\Big)^{1-\theta} &\leq c \cdot \min_{y \in X^*} \frac{\langle \nabla f(x); x - x^* \rangle}{||x - x^*||} \\ &\leq c\, ||\nabla f(x)||\,.\end{aligned}$$

Note that with $\theta = 1/2$, this is the sufficient condition [Garber and Hazan, 2015, (2)] implied by strong convexity that leads to $\mathcal{O}(1/T^2)$ convergence rates. Hence combining both we recover this recursive inequality for $h_t = f(x_t) - f(x^*)$

$$h_{t+1} \leq h_t \cdot \max\left\{\frac{1}{2}, 1 - \frac{\alpha}{8Lc} h_t^{1-\theta}\right\}.$$

When $\theta = 0$ (convexity), this leads to the classical $\mathcal{O}(1/T)$ rate. When $\theta = 1/2$ the above recursion leads to a $\mathcal{O}(1/T^2)$ rate as in [Garber and Hazan, 2015, proof of Theorem 2]. Then for any non-negative constants (k, C), such that $\frac{2-2^\beta}{2^\beta - 1} \leq k$ and $\max\{h_0 k^{1/\beta}, \frac{2}{\big((\beta - (1-\beta)(2^\beta - 1))M\big)^{1/\beta}}\} \leq C$ (with $M \triangleq \frac{\alpha}{8L}$), we have

$$h_t \leq \frac{C}{(t+1)^{1/(1-\theta)}}.$$

and the desired result. ■

Theorem 2.A.1 interpolates between the general $\mathcal{O}(1/T)$ rate for smooth convex functions and the $\mathcal{O}(1/T^2)$ rate for smooth and strongly convex functions.

2.B Analysis under Hölder Smoothness

In the following we generalize our results on convergence rates using a refined regularity assumption on f. A differentiable function f is (L, s)-Hölder smooth on $\mathcal{C}$ when

$$||\nabla f(x) - \nabla f(y)||_2 \leq L||x - y||_2^{s-1}, \quad \text{for } x, y \in \mathcal{C},$$

with $s \in]1, 2]$. Hölder smoothness interpolates between non-smooth ($s = 1$) and smooth ($s = 2$) assumptions. We write the analog of the away curvature (1.16) for (L, s)-Hölder smooth functions as

$$C_{f,s}^A \triangleq \sup_{\substack{x,u,v\in\mathcal{C} \\ \eta\in[0,1] \\ y=x+\eta(u-v)}} \frac{s}{\eta^s}(f(y) - f(x) - \eta\langle\nabla f(x), u - v\rangle).$$

Note that as in (1.16), f needs to be defined on the Minkowski sum $\mathcal{C}^A$. Let us now provide equivalent results for the complexity of Fractional Away-Step Frank-Wolfe algorithm and the complexity bound of the constant restart scheme with (L, s)-Hölder smooth functions.

Proposition 2.B.1 (Hölder Smooth Complexity)**.** *Let f be a (L, s)-Hölder smooth convex function with away curvature $C_{f,s}^A$ such that the r-strong-Wolfe primal bound in (2.10) holds on $\mathcal{C}$ with $\mu > 0$. Let $\gamma > 0$ and assume $x_0 \in \mathcal{C}$ is such that $e^{-\gamma} w(x_0, \mathcal{S}_0)/2 \leq C_{f,s}^A$. Algorithm 5 outputs an iterate $x_T \in \mathcal{C}$ such that*

$$w(x_T, \mathcal{S}_T) \leq w(x_0, \mathcal{S}_0)e^{-\gamma}$$

after at most (with $r = \frac{1}{1-\theta}$)

$$T \leq |\mathcal{S}_0| - |\mathcal{S}_T| + 2^{1+\frac{s}{s-1}} \frac{s}{s-1} e^{\frac{s}{s-1}\gamma} \mu (C_{f,s}^A)^{\frac{1}{s-1}} w(x_0, \mathcal{S}_0)^{r-\frac{s}{s-1}}$$

iterations, where $\mathcal{S}_0$ and $\mathcal{S}_T$ are the supports of respectively x_0 and x_T.

Proof. The proof is very similar to that required for smooth-functions, so we only detail key points. The update direction satisfies

$$r_t^T d_t > e^{-\gamma} w_0/2 \ .$$

Applying the definition of the Hölder curvature

$$f(x_t) - f(x_t + \eta d_t) \geq \max_{\eta\in[0,\eta_{\max}]} \{\eta e^{-\gamma} w_0/2 - \frac{\eta^s}{s} C_{f,s}^A\} = \max_{\eta\in[0,\eta_{\max}]} g(\eta) \ .$$

The unconstrained maximum of g is reached at $\eta^* = \left(e^{-\gamma} \frac{w_0}{2C_{f,s}^A}\right)^{\frac{1}{s-1}}$. Hence with the burn-in phase hypothesis, we guarantee $\eta^* \leq 1$. With classical arguments, for all non-drop steps, the progress in the objective function value is lower bounded by

$$f(x_t) - f(x_t + \eta d_t) \geq \frac{1}{(C_{f,s}^A)^{\frac{1}{s-1}}} \frac{s-1}{s} 2^{-\frac{s}{s-1}} e^{-\gamma\frac{s}{s-1}} w_0^{\frac{s}{s-1}}.$$

It finally follows that

$$T \leq 2\mu w_0^{r-\frac{s}{s-1}} 2^{\frac{s}{s-1}} \frac{s}{s-1} e^{\gamma \frac{s}{s-1}} + |\mathcal{S}_0| - |\mathcal{S}_T|$$

which is the desired bound. ■

We are ready to establish the convergence rates of our restart scheme in the Hölder smooth case.

Theorem 2.B.2 (Hölder rate for constant restart schemes)**.** *Let f be a (L,s)-Hölder smooth convex function with Hölder curvature $C_{f,s}^A$, satisfying (θ, c)-HEB on $\mathcal{C}$, and $\mathcal{C}$ satisfying a δ-Scaling inequality. Let $\gamma > 0$ and assume $x_0 \in K$ is such that $e^{-\gamma} w(x_0, \mathcal{S}_0)/2 \leq C_{f,s}^A$. With $\gamma_k = \gamma$, the output of Algorithm 6 satisfies*

$$f(x_T) - f^* \leq w_0 \frac{1}{\left(1 + \tilde{T} C_\gamma^\tau\right)^{\frac{1}{\tau}}} \quad \text{when } 1 \leq r < \frac{s}{s-1}$$

after T steps, with $w_0 \triangleq w(x_0, \mathcal{S}_0)$, $\tilde{T} \triangleq T - (|\mathcal{S}_0| - |\mathcal{S}_T|)$, and $\tau \triangleq \frac{s}{s-1} - r$. Also

$$C_\gamma^\tau \triangleq \frac{e^{\gamma\tau} - 1}{C_s e^{\frac{s}{s-1}\gamma} w(x_0)^\tau},$$

with $C_s \triangleq 2^{1+\frac{s}{s-1}} \frac{s}{s-1} \frac{c}{\delta} (C_{f,s}^A)^{\frac{1}{s-1}}$.

Proof. Denote by R the number of restarts after T total inner iterations. We get

$$T \leq \sum_{i=0}^{R-1} |\mathcal{S}_i| - |\mathcal{S}_{i+1}| + 2^{1+\frac{s}{s-1}} \frac{s}{s-1} e^{\frac{s}{s-1}\gamma} (C_{f,s}^A)^{\frac{1}{s-1}} \mu w(x_i, \mathcal{S}_i)^{r-\frac{s}{s-1}} .$$

Since $w(x_i, \mathcal{S}_i) \leq w_0 e^{-\gamma i}$, it follows that

$$T \leq |\mathcal{S}_0| - |\mathcal{S}_T| + 2^{1+\frac{s}{s-1}} \frac{s}{s-1} e^{\frac{s}{s-1}\gamma} (C_{f,s}^A)^{\frac{1}{s-1}} \mu w_0^{r-\frac{s}{s-1}} \sum_{i=0}^{R-1} e^{-\gamma i (r - \frac{s}{s-1})} .$$

Write $C_s = 2^{1+\frac{s}{s-1}} \frac{s}{s-1} (C_{f,s}^A)^{\frac{1}{s-1}} \mu$ and $\tau = \frac{s}{s-1} - r$ we have

$$T \leq |\mathcal{S}_0| - |\mathcal{S}_T| + C_s e^{\frac{s}{s-1}\gamma} w_0^{r-\frac{s}{s-1}} \frac{e^{\gamma R \tau} - 1}{e^{\gamma\tau} - 1},$$

it follows that

$$e^{-\gamma R} \leq \frac{1}{\left(1 + (T - (|\mathcal{S}_0| - |\mathcal{S}_T|)) \frac{(e^{\gamma\tau} - 1)}{C_s e^{\frac{s}{s-1}\gamma} w(x_0)^\tau}\right)^{\frac{1}{\tau}}}.$$

which yields the desired result. ■

Note that $r < \frac{s}{s-1}$ is always ensured because $s \in]1, 2]$. In particular we only get linear convergence when $r = s = 2$ as for gradient methods [Roulet and d'Aspremont, 2017]. We now show, as in Proposition 2.3.2, that the assumption $e^{-\gamma} w(x_0, \mathcal{S}_0)/2 \leq C_f^A$ has a marginal impact on complexity when the function is (L,s)-Hölder smooth.

Proposition 2.B.3 (Burn-in phase for Hölder smooth functions)**.** *After at most*

$$4\frac{s}{s-1}\frac{e^\gamma}{\gamma}\ln\left(\frac{w_0}{2C^A_{f,s}}\right)+|\mathcal{S}_0|$$

cumulative iterations of Algorithm 5, with constant schedule parameter $\gamma>0$, we get a point x such that $e^{-\gamma}w(x,\mathcal{S})/2\leq C^A_{f,s}$ when f is (L,s)-Hölder smooth with $s>1$.

Proof. Assume we have $e^{-\gamma}w_0/2>C^A_{f,s}$. Classically, the curvature argument ensures that we have for non-drop steps

$$\begin{aligned} f(x_t)-f(x_{t+1}) &\geq \eta_t e^{-\gamma}w_0/2-\frac{\eta_t^s}{s}C^A_{f,s}\\ &\geq e^{-\gamma}w_0/2(1-1/s). \end{aligned}$$

Besides, T_f being the number of full steps and T the number of iterations before Fractional Away Frank-Wolfe stops,

$$f(x_0)-f(x_T)\geq T_f e^{-\gamma}w_0/2(1-1/s).$$

Combining this with $f(x_0)-f(x_T)\leq f(x_0)-f(x^*)\leq w_0$ we get

$$T_f\leq 2e^\gamma\frac{s}{s-1}.$$

Finally with the classical counting argument on drop steps, we obtain

$$T\leq 4e^\gamma\frac{s}{s-1}+|\mathcal{S}_0|-|\mathcal{S}_T|\ .$$

Denote R the number of calls to Fractional Away Frank-Wolfe before the last output $\hat{x}_R$ satisfies $e^{-\gamma}w(\hat{x},\mathcal{S}_{\hat{x}})/2>C^A_{f,s}$. The strong-Wolfe gap of the N^{th} output of Fractional Away Frank-Wolfe satisfies by definition

$$w(\hat{x}_N)\leq e^{-N\gamma}w_0,$$

hence we have

$$R\leq\frac{1}{\gamma}\ln\left(\frac{w_0}{2C^A_{f,s}}\right)\ .$$

Finally each round of Fractional Away Frank-Wolfe under the initial assumption that $e^{-\gamma}w(\hat{x}_i,\mathcal{S}_{\hat{x}_i})/2>C^A_{f,s}$ require at most $4e^\gamma\frac{s}{s-1}+|\mathcal{S}_{\hat{x}_i}|-|\mathcal{S}_{\hat{x}_{i+1}}|$ iterations. Hence a total T_t of

$$\begin{aligned} T_t &\leq \sum_{i=1}^{R}4e^\gamma\frac{s}{s-1}+|\mathcal{S}_{\hat{x}_i}|-|\mathcal{S}_{\hat{x}_{i+1}}|\\ &\leq 4Re^\gamma\frac{s}{s-1}+|\mathcal{S}_0|\\ &\leq 4\frac{s}{s-1}\frac{e^\gamma}{\gamma}\ln\left(\frac{w_0}{2C^A_{f,s}}\right)+|\mathcal{S}_0| \end{aligned}$$

which is the desired result. ∎

2.C One Shot Application of the Fractional Away-Step Frank-Wolfe

Running once Fractional Away-step Frank-Wolfe with a large value of γ allows to find an approximate minimizer with the desired precision. The following lemma explains the rate of convergence. Importantly the rate does not depend on r. Hence there is no hope of observing linear convergence for the strongly convex case.

Lemma 2.C.1. *Let f be a smooth convex function, $\epsilon > 0$ be a target accuracy, and $x_0 \in \mathcal{C}$ be an initial point. Then for any $\gamma > \ln \frac{w(x_0)}{\epsilon}$, Algorithm 5 satisfies:*

$$f(x_T) - f^* \leq \epsilon,$$

for $T \geq \frac{2C_f^{\mathcal{A}}}{\epsilon}$.

Proof. We can stop the algorithm as soon as the criterion $w(x_t) < \epsilon$ in step 2 is met or we observe an away step, whichever comes first. In former case we have $f(x_t) - f^* \leq w(t) < \epsilon$, in the latter it holds

$$f(x_t) - f^* \leq -\nabla f(x_t)(d_t^{FW}) \leq \epsilon/2 < \epsilon.$$

Thus, when the algorithms stops, we have achieved the target accuracy and it suffices to bound the number of iterations required to achieve that accuracy. Moreover, while running, the algorithm only executes Frank-Wolfe and we drop the FW superscript in the directions; otherwise we would have stopped.

From the proof of Proposition 2.3.1, we have each Frank-Wolfe step ensures progress of the form

$$f(x_t) - f(x_{t+1}) \geq \begin{cases} \frac{\langle r_t; d_t \rangle^2}{2C_f^{\mathcal{A}}} & \text{if } \langle r_t; d_t \rangle \leq C_f^{\mathcal{A}} \\ \langle r_t; d_t \rangle - C_f^{\mathcal{A}}/2 & \text{otherwise.} \end{cases}$$

For convenience, let $h_t \triangleq f(x_t) - f^*$. By convexity we have $h_t \leq \langle r_t; d_t \rangle$, so that the above becomes

$$f(x_t) - f(x_{t+1}) \geq \begin{cases} \frac{h_t^2}{2C_f^{\mathcal{A}}} & \text{if } h_t \leq C_f^{\mathcal{A}} \\ h_t - C_f^{\mathcal{A}}/2 & \text{otherwise.} \end{cases},$$

and moreover observe that the second case can only happen in the very first step: $h_1 \leq h_0 - (h_0 - C_f^{\mathcal{A}}/2) = C_f^{\mathcal{A}}/2 \leq 2C_f^{\mathcal{A}}/t$ for $t = 1$ providing the start of the following induction: we claim $h_t \leq \frac{2C_f^{\mathcal{A}}}{t}$.

Suppose we have established the bound for t, then for $t + 1$, we have

$$h_{t+1} \leq \left(1 - \frac{h_t}{2C_f^{\mathcal{A}}}\right) h_t \leq \frac{2C_f^{\mathcal{A}}}{t} - \frac{2C_f^{\mathcal{A}}}{t^2} \leq \frac{2C_f^{\mathcal{A}}}{t+1}.$$

The induction is complete and it follows that the algorithm requires $T \geq \frac{2C_f^{\mathcal{A}}}{\epsilon}$ to reach ϵ-accuracy. ■

Chapter 3

Frank-Wolfe on Uniformly Convex Sets

In Chapter 2, we investigated how relaxing properties of the objective function modifies the convergence rates of Frank-Wolfe algorithms. Here we focus on identifying which structures of the constraint sets $\mathcal{C}$ lead to accelerated rates with respect to the general $\mathcal{O}(1/T)$ for compact convex sets. In particular, we will not seek to relax any assumption related to the objective function.

In Chapter 1, we recalled that the original Frank-Wolfe method solves smooth constrained convex optimization problems at a generic sublinear rate of $\mathcal{O}(1/T)$. It enjoys accelerated convergence rates for two fundamental classes of constraints: polytopes and strongly-convex sets. Uniformly convex sets non-trivially subsume strongly convex sets and form a large variety of *curved* convex sets commonly encountered in machine learning and signal processing. For instance, the ℓ_p balls are uniformly convex for all $p > 1$, but strongly convex for $p \in]1, 2]$ only. In this chapter, we show that these sets induce accelerated convergence rates for the Frank-Wolfe algorithm, which continuously interpolate between known rates. Our accelerated convergence rates emphasize that it is the curvature of the constraint sets – not just their strong convexity – that leads to accelerated convergence rates for the Frank-Wolfe algorithm. These results also importantly highlight that the Frank-Wolfe algorithm is adaptive to much more generic constraint set structures, thus explaining faster empirical convergence. Finally, we also show accelerated convergence rates when the set is only locally uniformly convex and provide similar results in online linear optimization.

Contents

3.1 Introduction

The Frank-Wolfe method [Frank and Wolfe, 1956] (Algorithm 9) is a projection-free algorithm designed to solve

$$\operatorname*{argmin}_{x \in \mathcal{C}} f(x), \tag{OPT}$$

where $\mathcal{C}$ is a compact convex set and f a smooth convex function. Many recent algorithmic developments in this family of methods are motivated by appealing properties already contained in the original Frank-Wolfe algorithm. Each iteration requires to solve a Linear Minimization Oracle (see line 2 in Algorithm 9), instead of a projection or proximal operation that is not computationally competitive in various settings. Also, the Frank-Wolfe iterates are convex combinations of extreme points of $\mathcal{C}$, the solutions of the Linear Minimization Oracle. Hence, depending on the extremal structure of $\mathcal{C}$, early iterates may have a specific structure, being, *e.g.* , sparse or low rank for instance, that could be traded-off with the iterate approximation quality of problem (OPT). These fundamental properties are among the main features that contribute to the recent revival and extensions of the Frank-Wolfe algorithm [Clarkson, 2010b, Jaggi, 2011] used for instance in large-scale structured prediction [Bojanowski et al., 2014, 2015, Alayrac et al., 2016, Seguin et al., 2016, Miech et al., 2017, Peyre et al., 2017, Miech et al., 2018], quadrature rules in RKHS [Bach et al., 2012, Lacoste-Julien et al., 2015, Futami et al., 2019], optimal transport [Courty et al., 2016, Vayer et al., 2018, Paty and Cuturi, 2019, Luise et al., 2019], and many others.

Algorithm 9 Frank-Wolfe Algorithm

Input: $x_0 \in \mathcal{C}$, L upper bound on the Lipschitz constant.
1: **for** $t = 0, 1, \ldots, T$ **do**
2: $\quad v_t \in \operatorname*{argmax}_{v \in \mathcal{C}} \langle -\nabla f(x_t); v - x_t \rangle$ ▷ Linear minimization oracle
3: $\quad \gamma_t = \operatorname*{argmin}_{\gamma \in [0,1]} \gamma \langle v_t - x_t; \nabla f(x_t) \rangle + \frac{\gamma^2}{2} L ||v_t - x_t||^2$ ▷ Short step
4: $\quad x_{t+1} = (1 - \gamma_t) x_t + \gamma_t v_t$ ▷ Convex update
5: **end for**

Uniform Convexity. Uniform convexity is a global quantification of the curvature of a convex set $\mathcal{C}$. There exists several definitions, see for instance, [Goncharov and Ivanov, 2017, Theorem 2.1.] and [Abernethy et al., 2018, Molinaro, 2020] for the strongly convex case. Here, we focus on the generalization of a classic definition of the strong convexity of a set [Garber and Hazan, 2015].

Definition 3.1.1 (γ uniform convexity of $\mathcal{C}$). *A closed set $\mathcal{C} \subset \mathbb{R}^d$ is $\gamma_{\mathcal{C}}$-uniformly convex with respect to a norm $||\cdot||$, if for any $x, y \in \mathcal{C}$, any $\eta \in [0,1]$ and any $z \in \mathbb{R}^d$ with $||z|| = 1$, we have*

$$\eta x + (1-\eta)y + \eta(1-\eta)\gamma_{\mathcal{C}}(||x-y||)z \in \mathcal{C},$$

where $\gamma_{\mathcal{C}}(\cdot) \geq 0$ is a non-decreasing function. In particular when there exists $\alpha > 0$ and $q > 0$ such that $\gamma_{\mathcal{C}}(r) \geq \alpha r^q$, we say that $\mathcal{C}$ is (α, q)-uniformly convex or q-uniformly convex.

The uniform convexity assumption strengthens the convexity property of $\mathcal{C}$ that any line segment between two points is included in $\mathcal{C}$. It requires a scaled unit ball to fit in $\mathcal{C}$ and results in curved sets. Strongly convex sets are uniformly convex sets for which $\gamma_{\mathcal{C}}(r) \geq \alpha r^2$, *i.e.* $(\alpha, 2)$-uniformly convex sets. Two common families of uniformly convex sets are the ℓ_p-balls and p-Schatten balls which are uniformly convex for any $p > 1$ but strongly convex for $p \in]1, 2]$ only, *i.e.* 2-uniformly convex sets for $p \in]1, 2]$.

Convergence Rates for Frank-Wolfe. The Frank-Wolfe algorithm admits a tight [Canon and Cullum, 1968, Jaggi, 2013, Lan, 2013] general sublinear convergence rate of $\mathcal{O}(1/T)$ when $\mathcal{C}$ is a compact convex set and f is a convex L-smooth function. However, when the constraint set $\mathcal{C}$ is strongly-convex and $\inf_{x \in \mathcal{C}} ||\nabla f(x)|| > 0$, Algorithm 9 enjoys a linear convergence rate [Levitin and Polyak, 1966, Demyanov and Rubinov, 1970]. Later on, the work of [Dunn, 1979] showed that linear rates are maintained when the constraint set satisfies a condition subsuming local strong-convexity. Interestingly, this linear convergence regime *does not* require the strong-convexity of f, *i.e.* the lower quadratic additional structure comes from the constraint set rather than from the function. When x^* is in the interior of $\mathcal{C}$ and f is strongly convex, Algorithm 9 also enjoys a linear convergence rate [Guélat and Marcotte, 1986].

These two linear convergence regimes can both become arbitrarily bad as x^* gets close to the border of $\mathcal{C}$, and do not apply in the limit case where the unconstrained optimum of f lies at the boundary of $\mathcal{C}$. In this scenario, when the constraint set is strongly convex, Garber and Hazan [2015] prove a general sublinear rate of $\mathcal{O}(1/T^2)$ when f is L-smooth and μ-strongly convex. In early iterations, these convergence rates can beat badly-conditioned linear rates.

Other structural assumptions are known to lead to accelerated convergence rates. However, these require elaborate algorithmic enhancements of the original Frank-Wolfe algorithm. Polytopes received much attention in particular, with *corrective* or *away* algorithmic mechanisms [Guélat and Marcotte, 1986, Hearn et al., 1987] that lead to linear convergence rates under appropriate structures of the objective function [Garber and Hazan, 2013a, Lacoste-Julien and Jaggi, 2013, 2015b, Beck and Shtern, 2017, Gutman and Pena, 2018, Pena and Rodriguez, 2018]. Accelerated versions of Frank-Wolfe, when the constraint set is a trace-norm ball (a.k.a. nuclear balls) – which are neither polyhedral nor strongly convex [So, 1990] – have also received a lot of attention [Freund et al., 2017, Allen-Zhu et al., 2017, Garber et al., 2018] and are especially useful in matrix completion [Jaggi et al., 2010, Shalev-Shwartz et al., 2011, Harchaoui et al., 2012, Dudik et al., 2012].

Contributions. We show accelerated sublinear convergence rates for the Frank-Wolfe algorithm, with appropriate line-search, for smooth constrained optimization problems when the constraint set is globally or locally uniformly convex. These bounds generalize the rates of [Polyak, 1966, Demyanov and Rubinov, 1970], [Dunn, 1979], and [Garber and Hazan, 2015]

in their respective settings and fill the gap between all known convergence rates, *i.e.* between $\mathcal{O}(1/T)$ and the linear rate of [Levitin and Polyak, 1966, Demyanov and Rubinov, 1970, Dunn, 1979], and between $\mathcal{O}(1/T)$ and the $\mathcal{O}(1/T^2)$ rate of [Garber and Hazan, 2015] (see *e.g.* concluding remarks of [Garber and Hazan, 2015]). We also provide similar arguments that interpolate between known regret bounds in an example of projection-free online learning. Overall, we illustrate another key aspect of the Frank-Wolfe algorithms: they are adaptive to many generic structural assumptions.

Outline. In Section 3.2, we analyze the complexity of the Frank-Wolfe algorithm when the constraint set is uniformly convex, under various assumptions on f. In Section 3.2.3, we also establish accelerated convergence rate under weaker assumptions than global or local uniform convexity of the constraint set. In Section 3.3, we focus on the online optimization setting and provide analogous results to the previous section in term of regret bounds. In Section 3.4, we give some examples of uniformly convex sets and relate the uniform convexity notion for sets with that of spaces and functions.

Notation. We use d for the *ambient dimension* of the compact convex sets $\mathcal{C}$. We denote the *boundary* of $\mathcal{C}$ by $\partial\mathcal{C}$ and let $N_{\mathcal{C}}(x) \triangleq \{d \mid \langle d; y - x\rangle \leq 0,\ \forall y \in \mathcal{C}\}$ denote the *normal cone* at x with respect to $\mathcal{C}$. In the following, x^* is an (optimal) solution to (OPT) and (α, q) denotes the uniform convexity parameters of a set. p stands for the parameters for the various norm balls and might differ from q. We sometimes assume strict convexity of f for the sake of exposition (only). Given a norm $||\cdot||$ we denote by $||d||_* \triangleq \max_{||x||\leq 1}\langle x; d\rangle$ its dual norm and we let $h_t \triangleq f(x_t) - f(x^*)$ denote the primal gap.

3.2 Frank-Wolfe Convergence Analysis with Uniformly Convex Constraints

In Theorem 3.2.2, we show accelerated convergence rate of the Frank-Wolfe algorithm when the constraint set $\mathcal{C}$ is (α, q)-uniformly convex (with $q \geq 2$) and the smooth convex function satisfies $\inf_{x\in\mathcal{C}}||\nabla f(x)|| > 0$; this is the interesting case. In Section 3.2.3, we then explore *localized* uniform convexity on the set $\mathcal{C}$ and provide convergence rates in Theorem 3.2.5. In Theorem 3.2.10 we show that (α, q)-uniform convexity ensures convergence rates of the Frank-Wolfe algorithms in between the $\mathcal{O}(1/T)$ and $\mathcal{O}(1/T^2)$ [Garber and Hazan, 2015] when the function is strongly convex (and L-smooth), or satisfies a quadratic error bound at x^*. We also provide generalized convergence rates assuming Hölderian Error Bounds on f. In all of these scenarios, when the set is uniformly convex, the Frank-Wolfe algorithm (with short step) enjoys accelerated convergence rates with respect to $\mathcal{O}(1/T)$.

Proof Sketch. We now provide an informal discussion as to why the uniform convexity of $\mathcal{C}$ leads to accelerated convergence rates under the classical assumptions that $\inf_{x\in\mathcal{C}}||\nabla f(x)|| > 0$ and hence $x^* \in \partial\mathcal{C}$. Formal arguments are developed in the proof of Theorem 3.2.2. The key point is that if $\mathcal{C}$ is curved around x^* and f is L-smooth, when $||x_t - x^*||$ converges to zero, the quantity $||x_t - v_t||$ also converges to zero, which is generally not the case, for instance when the constraint set is a polytope.

In Figure 3.1 we show various such behaviors. Applying the L-smoothness of f to the Frank-Wolfe iterates, the classical iteration inequality is of the form (with $\gamma \in [0,1]$)

$$f(x_{t+1}) - f(x^*) \leq f(x_t) - f(x^*) - \gamma\langle -\nabla f(x_t); v_t - x_t\rangle + \frac{\gamma^2}{2} L||x_t - v_t||^2. \tag{3.1}$$

The non-negative quantity $\langle -\nabla f(x_t); v_t - x_t\rangle$ participates in guaranteeing the function decrease, counter-balanced with $||x_t - v_t||^2$. The convergence rate then depends on specific relative quantification of these various terms, that we call scaling inequalities in Lemma 3.2.1 and 3.2.4.

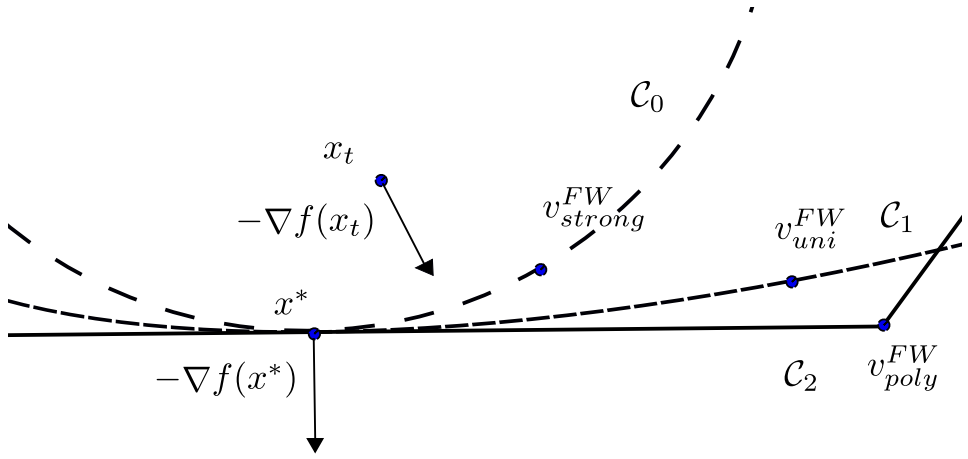


Figure 3.1: v_{strong}^{FW}, v_{uni}^{FW}, v_{poly}^{FW} represent the various FW vertices from the strongly convex set $\mathcal{C}_0$, the uniformly convex set $\mathcal{C}_1$ and the polytope $\mathcal{C}_2$.

3.2.1 Scaling Inequality on Uniformly Convex Sets

The following lemma outlines that the uniform convexity of $\mathcal{C}$ implies an upper bound on the distance between the current iterate and the Frank-Wolfe vertex as a power of the Frank-Wolfe gap. Note that the uniform convexity is defined with respect to any norm, and not just in terms of an Hilbertian structure. To be even more generic, the uniform convexity can be defined with respect to gauge functions that are not necessarily norms, see, for instance, the strong-convexity of [Molinaro, 2020].

Lemma 3.2.1. *Assume the compact $\mathcal{C} \subset \mathbb{R}^d$ is an (α, q)-uniformly convex set with respect to a norm $||\cdot||$, with $\alpha > 0$ and $q \geq 2$. Consider $x \in \mathcal{C}$, $\phi \in \mathbb{R}^d$ and $v_\phi \in \operatorname{argmax}_{v\in\mathcal{C}} \langle \phi; v\rangle$. Then, we have $\langle \phi; v_\phi - x\rangle \geq \frac{\alpha}{2}||v_\phi - x||^q||\phi||_*$. In particular for an iterate x_t and its associated Frank-Wolfe vertex v_t, this yields*

$$\langle -\nabla f(x_t); v_t - x_t\rangle \geq \frac{\alpha}{2}||v_t - x_t||^q||\nabla f(x_t)||_*. \tag{Global-Scaling}$$

Proof. Because $\mathcal{C}$ is (α, q)-uniformly convex, we have that for any $z \in \mathbb{R}^d$ of unit norm $(x + v_\phi)/2 + \alpha/4||x - v_\phi||^q z \in \mathcal{C}$. By optimality of v_ϕ, we have $\langle \phi; v_\phi\rangle \geq \langle \phi; (x + v_\phi)/2\rangle + \alpha/4||x - v_\phi||^q\langle \phi, z\rangle$. Hence, choosing the best z implies $\langle \phi; v_\phi - x\rangle \geq \alpha/2||v_\phi - x||^q||\phi||_*$. ∎

In other words, when $\mathcal{C}$ is uniformly convex, (Global-Scaling) quantifies the trade-off between the Frank-Wolfe gap $g(x_t) \triangleq \langle \nabla f(x_t); x_t - v_t\rangle$ and the value of $||x_t - v_t||$ under consideration in (3.1).

3.2.2 Interpolating linear and sublinear rates

To our knowledge, no accelerated convergence rate of the Frank-Wolfe algorithm is known when the constraint set is uniformly convex but not strongly convex. We fill this gap in Theorem 3.2.2 below. When q goes to $+\infty$, we recover the classic sublinear convergence rate of $\mathcal{O}(1/T)$.

Theorem 3.2.2. *Consider a convex L-smooth function f and a compact convex set $\mathcal{C}$. Assume that $\mathcal{C}$ is (α, q)-uniformly convex set with respect to a norm $||\cdot||$, with $q \geq 2$. Assume $||\nabla f(x)||_* \geq c > 0$ for all $x \in \mathcal{C}$. Then the iterates of the Frank-Wolfe algorithm, with short step as in Line 3 of Algorithm 9 or exact line search, satisfy*

$$\begin{cases} f(x_T) - f(x^*) \leq M/(T+k)^{1/(1-2/q)} & \text{when } q > 2 \\ f(x_T) - f(x^*) \leq (1-\rho)^T h_0 & \text{when } q = 2, \end{cases} \tag{3.2}$$

with $\rho = max\{\frac{1}{2}, 1 - c\alpha/L\}$, $k \triangleq (2-2^\eta)/(2^\eta - 1)$ and $M \triangleq max\{h_0 k^{1/\eta}, 2/((\eta - (1-\eta)(2^\eta - 1))C)^{1/\eta}\}$, where $\eta \triangleq 1 - 2/q$ and $C \triangleq (c\alpha/2)^{2/q}/(2L)$.

Proof. By L-smoothness of f and because of the short step, we have for $\gamma \in [0,1]$

$$f(x_{t+1}) \leq f(x_t) - \gamma g(x_t) + \frac{\gamma^2}{2} L ||x_t - v_t||^2,$$

where $g(x_t)$ is the Frank-Wolfe gap. With $\gamma = \min\{1, g(x_t)/(L||x_t - v_t||^2)\}$ we have

$$f(x_{t+1}) \leq f(x_t) - \frac{g(x_t)}{2} \cdot \min\Big\{1; \frac{g(x_t)}{L||x_t - v_t||^2}\Big\}.$$

Applying Lemma 3.2.1 with $\phi = -\nabla f(x_t)$ gives $g(x_t) \geq \alpha/2 ||x_t - v_t||^q ||\nabla f(x_t)||_*$. Then

$$\frac{g(x_t)}{||x_t - v_t||^2} = \Big(\frac{g(x_t)^{q/2-1} g(x_t)}{||x_t - v_t||^q}\Big)^{2/q} \geq \Big(\alpha/2 ||\nabla f(x_t)||_*\Big)^{2/q} g(x_t)^{1-2/q}. \tag{3.3}$$

Finally, because $g(x_t) \geq f(x_t) - f(x^*) = h(x_t)$, we have

$$h(x_{t+1}) \leq h(x_t) - \frac{h(x_t)}{2} \min\Big\{1; \Big(\alpha/2 ||\nabla f(x_t)||_*\Big)^{2/q} h(x_t)^{1-2/q}/L\Big\},$$

and hence

$$h(x_{t+1}) \leq h(x_t) \cdot \max\Big\{\frac{1}{2}; 1 - \Big(\alpha/2 ||\nabla f(x_t)||_*\Big)^{2/q} h(x_t)^{1-2/q}/(2L)\Big\}. \tag{3.4}$$

Then, by assumption, for all $x \in \mathcal{C}$, we have $||\nabla f(x)||_* > c > 0$ and hence (3.4) becomes

$$h(x_{t+1}) \leq h(x_t) \cdot \max\Big\{\frac{1}{2}; 1 - (c\alpha/2)^{2/q} h(x_t)^{1-2/q}/(2L)\Big\}.$$

We solve the recursion with Lemma 3.A.1; when $q = 2$ we recover the linear convergence rate. ■

Remark 3.2.3. *The convergence rates in Theorem 3.2.2 imply convergence rates in term of distance to optimum by applying Lemma 3.2.1 with $\phi = -\nabla f(x^*)$ and convexity of f. Indeed, this yields*

$$||x_t - x^*||^q \leq \frac{2}{c\alpha} \langle -\nabla f(x^*); x^* - x_t \rangle \leq \frac{2}{c\alpha} (f(x_t) - f(x^*)).$$

Hence, to obtain convergence rates in terms of the distance of the iterates to the optimum, the uniform convexity of the set supersedes that of the function, which is not needed here.

3.2.3 Convergence Rates with Local Uniform Convexity

Theorem 3.2.2 relies on the global uniform convexity of the set. Actually, for the strongly convex case, it is equivalent to the global scaling inequality (Global-Scaling), see [Goncharov and Ivanov, 2017, Theorem 2.1 (g)]. However, weaker assumptions also lead to accelerated convergence rates of the Frank-Wolfe algorithm. In Theorem 3.2.5, we show accelerated convergence rates assuming a *local* scaling inequality at x^*. We then study the sets for which such an inequality holds. We say that a local scaling inequality holds at $x^* \in \mathcal{C}$, when there exists an $\alpha > 0$ and $q \geq 2$ such that for all $x \in \mathcal{C}$

$$\langle -\nabla f(x^*); x^* - x \rangle \geq \alpha/2 ||\nabla f(x^*)||_* \cdot ||x^* - x||^q. \tag{Local-Scaling}$$

This combines the position of $-\nabla f(x^*)$ with respect to the normal cone of $\mathcal{C}$ at x^* and the local geometry of $\mathcal{C}$ at x^*, see Remark 3.2.8. When the set $\mathcal{C}$ is globally (α, q)-uniformly convex, this is a direct consequence of Lemma 3.2.1 because $-\nabla f(x^*) \in N_{\mathcal{C}}(x^*)$. In the following lemma, we prove that it is also a consequence of a natural definition of local uniform convexity of $\mathcal{C}$ at x^*.

Lemma 3.2.4. *Consider a compact convex set $\mathcal{C}$ and x^* a solution to (OPT). Assume that $\mathcal{C}$ is locally (α, q)-uniformly convex at x^* with respect to $||\cdot||$ in the sense that, for all $x \in \mathcal{C}$, $\eta \in [0,1]$ and unit norm $z \in \mathbb{R}^d$, we have $\eta x^* + (1-\eta)x + \eta(1-\eta)\alpha||x^* - x||^q z \in \mathcal{C}$. Then (Local-Scaling) holds at x^* with parameters (α, q).*

Proof. By definition of local uniform convexity between x^* and x, we have that for any $z \in \mathbb{R}^d$ of unit norm $(x^* + x)/2 + \alpha/4||x^* - x||^q z \in \mathcal{C}$. Then, by optimality of x^*, *i.e.* $x^* \in \operatorname{argmax}_{v \in \mathcal{C}} \langle -\nabla f(x^*); v \rangle$, we have $\langle -\nabla f(x^*); x^* \rangle \geq \langle -\nabla f(x^*); (x^* + x)/2 \rangle + \alpha/4||x^* - x||^q \langle -\nabla f(x^*), z \rangle$. Choosing the best z and subtracting both sides by $\langle -\nabla f(x^*); x \rangle$, implies

$$\langle -\nabla f(x^*); x^* - x \rangle \geq \alpha/2 ||x^* - x||^q ||\nabla f(x^*)||_*.$$

■

We obtain sublinear convergence rates that are systematically better than the $\mathcal{O}(1/T)$ baseline for any $q \geq 2$.

Theorem 3.2.5. *Consider f an L-smooth convex function and a compact convex set $\mathcal{C}$. Assume $||\nabla f(x)||_* > c > 0$ for all $x \in \mathcal{C}$ and write $x^* \in \partial\mathcal{C}$ a solution of (OPT). Further, assume that the convex set $\mathcal{C}$ satisfies a local scaling inequality at x^* with parameters (α, q). Then the iterates of the Frank-Wolfe algorithm, with short step satisfy*

$$\begin{cases} f(x_T) - f(x^*) \leq M/(T+k)^{\frac{1}{1-2/(q(q-1))}} & \text{when } q > 2 \\ f(x_T) - f(x^*) \leq (1-\rho)^T h_0 & \text{when } q = 2, \end{cases} \tag{3.5}$$

with $\rho = max\{\frac{1}{2}, 1 - c\alpha/L\}$, $k \triangleq (2 - 2^\eta)/(2^\eta - 1)$ and $M \triangleq max\{h_0 k^{1/\eta}, 2/((\eta - (1-\eta)(2^\eta - 1))C)^{1/\eta}\}$, where $\eta \triangleq 1 - 2/(q(q-1))$ and $C \triangleq 1/(2LH^2)$. Note that H depends only on C, α, L and q (see Lemma 3.2.7).

Remark 3.2.6. *When the local scaling inequality (Local-Scaling) holds with $q = 2$, we obtain the same linear convergence regime as in (3.2). With $q > 2$, the sublinear convergence rates are of order $\mathcal{O}(1/T^{1/(1-2/(q(q-1)))})$ instead of $\mathcal{O}(1/T^{1/(1-2/q)})$ when the set is (α, q)-uniformly convex and the global scaling inequality (Global-Scaling) holds. It is an open question to close this gap in the convergence regime with the local scaling inequality only.*

The local scaling inequality expresses a property between x^* and any $x \in \mathcal{C}$. In the following lemma, we show that albeit we only have access to a local scaling inequality, it is still possible to control the variation of the distance of the iterate to its Frank-Wolfe vertex $||x_t - v_t||$ in terms of a power of the primal gap, see beginning of Section 3.2 for a qualitative explanation. This is key for the proof of Theorem 3.2.5.

Lemma 3.2.7. *Consider f a L-smooth convex function and a compact convex set $\mathcal{C}$. Assume $inf_{x\in\mathcal{C}}||\nabla f(x)||_* > c > 0$ and write $x^* \in \partial\mathcal{C}$ the solution of* (OPT)*. Assume that $\mathcal{C}$ satisfies a local scaling inequality at x^* for problem* (OPT) *with $\alpha > 0$ and $q \geq 2$, i.e. for all $x \in \mathcal{C}$*

$$\langle -\nabla f(x^*); x^* - x\rangle \geq \alpha/2||\nabla f(x^*)||_* \cdot ||x^* - x||^q \tag{3.6}$$

Write $v_t \triangleq argmax_{v\in\mathcal{C}}\langle -\nabla f(x_t); v\rangle$ the Frank-Wolfe vertex. Assume that $h_t = f(x_t) - f(x^*) \leq 1$ (a simple burn-in phase). Then, we have

$$||x_t - v_t|| \leq H h_t^{1/(q(q-1))}, \tag{3.7}$$

with $H \triangleq 2 \cdot max\Big\{\Big(\frac{2L}{c\alpha}\Big)^{1/(q-1)}\Big(\frac{2}{c\alpha}\Big)^{1/(q(q-1))}, \Big(\frac{2}{c\alpha}\Big)^{1/q}\Big\}$.

Proof. We apply the local scaling inequality (3.6) with $x = v_t$ and $x = x_t$. We obtain two important inequalities: one that upper bounds $||x - x^*||$ in terms of $f(x) - f(x^*)$ and another that upper bounds $||v_t - x^*||$ in terms of $||x^* - x_t||$, where v_t is the Frank-Wolfe vertex related to iterate x_t. These two inequalities rely of convexity, L-smoothness and (3.6), but do not rely on strong convexity of the function f.

By optimality of the Frank-Wolfe vertex v_t, we have $\nabla f(x_t)^T v_t \leq \nabla f(x_t)^T x^*$. Hence, combining that with Cauchy-Schwartz, we get

$$\begin{aligned} ||\nabla f(x^*) - \nabla f(x_t)||\, ||v_t - x^*|| &\geq \langle \nabla f(x^*) - \nabla f(x_t); v_t - x^*\rangle + \underbrace{\langle \nabla f(x_t); v_t - x^*\rangle}_{\leq 0} \\ &\geq \langle \nabla f(x^*); v_t - x^*\rangle \geq c\alpha/2||v_t - x^*||^q. \end{aligned}$$

Then, L-smoothness applied to the left hand side leaves us with

$$||x_t - x^*|| \geq \frac{c\alpha}{2L}||v_t - x^*||^{q-1}, \tag{3.8}$$

and a triangular inequality gives

$$\begin{aligned} ||x_t - v_t|| &\leq ||v_t - x^*|| + ||x^* - x_t|| \\ ||x_t - v_t|| &\leq \Big(\frac{2L}{c\alpha}\Big)^{1/(q-1)}||x_t - x^*||^{1/(q-1)} + ||x^* - x_t||. \end{aligned}$$

Finally applying (3.6) with $x = x_t$ and using that $\inf_{x\in\mathcal{C}}||\nabla f(x)||_* > c > 0$, we have $||x_t - x^*|| \leq \Big(\frac{2}{c\alpha}\Big)^{1/q} h_t^{1/q}$ which leads to

$$||x_t - v_t|| \leq \Big(\frac{2L}{c\alpha}\Big)^{1/(q-1)}\Big(\frac{2}{c\alpha}\Big)^{1/(q(q-1))} h_t^{1/(q(q-1))} + \Big(\frac{2}{c\alpha}\Big)^{1/q} h_t^{1/q}\ .$$

We can simplify this previous expression, and we assumed without loss of generality (*i.e.* up to a burning-phase) that $h_t \leq 1$, which implies for $q \geq 2$ that $h_t^{1/(q(q-1))} \geq h_t^{1/q}$. With $H \triangleq 2 \cdot \max\Big\{\Big(\frac{2L}{c\alpha}\Big)^{1/(q-1)}\Big(\frac{2}{c\alpha}\Big)^{1/(q(q-1))}, \Big(\frac{2}{c\alpha}\Big)^{1/q}\Big\}$, we then have

$$||x_t - v_t|| \leq H h_t^{1/(q(q-1))}\ .$$

■

We now proceed with the proof of Theorem 3.2.5.

Proof of Theorem 3.2.5. With Lemma 3.2.7, which satisfies the assumption of Theorem 3.2.5, we have

$$||x_t - v_t|| \leq H h_t^{1/(q(q-1))},$$

with $H \triangleq 2 \cdot \max\Big\{\Big(\frac{2L}{c\alpha}\Big)^{1/(q-1)}\Big(\frac{2}{c\alpha}\Big)^{1/(q(q-1))}, \Big(\frac{2}{c\alpha}\Big)^{1/q}\Big\}$. We plug this last expression in the classical descent guarantee given by L-smoothness

$$\begin{aligned} h_{t+1} &\leq (1-\gamma)h_t + \gamma^2 \frac{L}{2}||v_t - x_t||^2 \\ h_{t+1} &\leq (1-\gamma)h_t + \gamma^2 \frac{L}{2} H^2 h_t^{2/(q(q-1))} \ . \end{aligned}$$

The optimal decrease $\gamma \in [0,1]$ is $\gamma^* = \min\Big\{\frac{h_t^{1-2/(q(q-1))}}{LH^2}, 1\Big\}$. When $\gamma^* = 1$, or equivalently $h_t \geq (LH^2)^{1-2/(q(q-1))}$, we have $h_{t+1} \leq h_t/2$. In other words, for the very first iterations, there is a brief linear convergence regime. Otherwise, when $\gamma^* \leq 1$, we have

$$h_{t+1} \leq h_t\Big(1 - \frac{1}{2LH^2} h_t^{1-2/(q(q-1))}\Big) \ . \tag{3.9}$$

When $q = 2$, this corresponds to the strongly convex case and we recover the classical linear-convergence regime. We conclude using Lemma 3.A.1 that the rate is $\mathcal{O}\Big(1/T^{1/(1-2/(q(q-1)))}\Big)$. ∎

A similar approach appears in [Dunn, 1979] which introduces the following functional

$$a_{x^*}(\sigma) \triangleq \inf_{\substack{x \in \mathcal{C} \\ ||x - x^*|| \geq \sigma}} \langle \nabla f(x^*); x - x^* \rangle,$$

and shows than when there exists $A > 0$ such that $a_{x^*}(\sigma) \geq A||x - x^*||^2$, then the Frank-Wolfe algorithm converges linearly, under appropriate line-search rules. This result of [Dunn, 1979] thus subsumes that of [Levitin and Polyak, 1966, Demyanov and Rubinov, 1970]. However, no analysis was conducted for uniformly (but not strongly) convex set.

In Lemma 3.2.4 we showed that a given quantification of local uniform convexity implies the local scaling inequality and hence accelerated convergence rates. However, there are many situations where such a local notion of uniform convexity does not hold but (Local-Scaling) does. This was the essence of [Dunn, 1979, Remark 3.5.] that we state here.

Corollary 3.2.8. *Assume there exists a compact and (α, q)-uniformly convex set Γ such that $\mathcal{C} \subset \Gamma$ and $N_\Gamma(x^*) \subset N_\mathcal{C}(x^*)$, where x^* is the solution of* (OPT)*. If $-\nabla f(x^*) \in N_\Gamma(x^*)$, then* (Local-Scaling) *holds at x^* with the (α, q) parameters.*

Proof. Here, because $N_\Gamma(x^*) \subset N_\mathcal{C}(x^*)$, we have that $x^* \in \operatorname{argmax}_{v \in \Gamma}\langle -\nabla f(x^*); v\rangle$. Also, for $x \in \mathcal{C} \subset \Gamma$, by (α, q)-uniform convexity of Γ, we also have that for any $z \in \mathbb{R}^d$ of unit norm that $(x^* + x)/2 + \alpha/4||x^* - x||^q z \in \Gamma$. Then, by optimality of x^*, we have $\langle -\nabla f(x^*); x^*\rangle \geq \langle -\nabla f(x^*); (x^* + x)/2\rangle + \alpha/4||x^* - x||^q \langle -\nabla f(x^*), z\rangle$. Choosing the best z and subtracting both sides by $\langle -\nabla f(x^*); x\rangle$, implies (for any $x \in \mathcal{C}$) $\langle -\nabla f(x^*); x^* - x\rangle \geq \alpha/2||x^* - x||^q ||\nabla f(x^*)||_*$. ∎

There exist numerous notions of local uniform convexity of a set that may imply local scaling inequalities. See for instance, the local directional strong convexity in [Goncharov and Ivanov, 2017, §Local Strong Convexity]. Alternatively, in the context of functions, Hölderian Errors Bounds (HEB) offer a weaker description of localized uniform convexity assumptions while retaining the same convergence rates [Kerdreux et al., 2019]. And these are known to hold generically for various classes of function [Lojasiewicz, 1965, Kurdyka, 1998, Bolte et al., 2007]. Obtaining a similar characterization for set is of interest. In particular, it is natural to relate enhanced convexity properties of the set gauge function $||\cdot||_{\mathcal{C}}$ [Rockafellar, 1970a, §15] to convexity properties of the set or directly to local scaling inequalities. For instance, local uniform convexity of the gauge $||\cdot||_{\mathcal{C}}$ implies a local scaling inequality for $\mathcal{C}$ (see Lemma 3.B.1). This suggests that error bounds as guaranteed with Łojasiewicz-type arguments on the gauge function should imply local scaling inequalities, showing that theses inequalities hold somewhat generically.

3.2.4 Interpolating Sublinear Rates for Arbitrary x^*

When the function is μ-strongly convex and the set $\mathcal{C}$ is α-strongly convex, Garber and Hazan [2015] show that the Frank-Wolfe algorithm (with short step) enjoys a general $\mathcal{O}(1/T^2)$ convergence rate. In particular, this result does not depend on the location of x^* with respect to $\mathcal{C}$. We now generalize this result by relaxing the strong convexity of the constraint set $\mathcal{C}$ and the quadratic error bound on f [Garber and Hazan, 2015, (1)].

Hölderian Error Bounds. Let f be a strictly convex L-smooth function and $x^* = \operatorname{argmin}_{x\in\mathcal{C}} f(x)$ where $\mathcal{C}$ is a compact convex set; the strict convexity assumption is only required to simplify exposition and the results hold more generally with the usual generalizations. We say that f satisfies a (μ, θ)-Hölderian Error Bound when there exists $\theta \in [0, 1/2]$ such that

$$||x - x^*|| \leq \mu(f(x) - f(x^*))^\theta. \tag{HEB}$$

When the function f is subanalytic, (HEB) is known to hold generically [Lojasiewicz, 1965, Kurdyka, 1998, Bolte et al., 2007]. For instance, when f is (μ, r)-uniformly convex with $r \geq 2$ (see Definition 3.D.1), then it satisfies a $((2/\mu)^{1/r}, 1/r)$-Hölderian Error Bound, which follows from

$$f(x_t) \geq f(x^*) + \underbrace{\langle \nabla f(x^*); x_t - x^* \rangle}_{\geq 0} + \frac{\mu}{2}||x_t - x^*||_2^r.$$

Hence we generalize the convergence result of [Garber and Hazan, 2015] and show that as soon as the set $\mathcal{C}$ is (α, q)-uniformly convex with $q \geq 2$ and the function f satisfies a non-trivial (μ, θ)-HEB, the Frank-Wolfe algorithm (with short step) enjoys an accelerated convergence rate with respect to $\mathcal{O}(1/T)$. In particular when f is μ-strongly convex, it satisfies a $(\mu, 1/2)$-HEB and by varying $q \geq 2$ we interpolate all sublinear convergence rates between $\mathcal{O}(1/T)$ and $\mathcal{O}(1/T^2)$.

In Lemma 3.2.9, we will show an upper bound on $||x_t - v_t||$ when combining the uniform convexity of $\mathcal{C}$ and a Hölderian Error Bound for f. Lemma 3.2.9 is then the basis for the convergence analysis and similar to Lemma 3.2.1. Overall, Theorem 3.2.2, Theorem 3.2.5 and Theorem 3.2.10 give an almost complete picture of all the accelerated convergence regimes one can expect with the vanilla Frank-Wolfe algorithm.

Lemma 3.2.9. *Consider a compact and (α, q)-uniformly convex set $\mathcal{C}$ with respect to $||\cdot||$. Denote f a strictly convex L-smooth function and $x^* = argmin_{x\in\mathcal{C}} f(x)$. Assume that f satisfies a (μ, θ)-Hölderian Error Bound $||x - x^*|| \leq \mu(f(x) - f(x^*))^\theta$ with $\theta \in [0, 1/2]$. Then for $x_t \in \mathcal{C}$ we have $\alpha/\mu||x_t - v_t||^q h_t^{1-\theta} \leq g(x_t)$, where $g(x_t)$ is the Frank-Wolfe gap and v_t the Frank-Wolfe vertex.*

Proof. By Lemma 3.2.1 we have $g(x_t) \geq \alpha||x_t - v_t||^q||\nabla f(x_t)||_*$. Then, by combining the convexity of f, Cauchy-Schwartz and (μ, θ)-Hölderian Error Bound, we have

$$f(x) - f(x^*) \leq \langle \nabla f(x); x - x^* \rangle \leq ||\nabla f(x)||_* \cdot ||x - x^*|| \leq \mu||\nabla f(x)||_* \cdot \big(f(x) - f(x^*)\big)^\theta,$$

so that $\big(f(x) - f(x^*)\big)^{1-\theta} \leq ||\nabla f(x)||_*$ and finally $g(x_t) \geq \alpha||x_t - v_t||^q h_t^{1-\theta}$. ∎

Theorem 3.2.10. *Consider a L-smooth convex function f that satisfies a (μ, θ)-HEB with $\mu > 0$ and $\theta \in]0, 1/2]$. Assume $\mathcal{C}$ is a compact and (α, q)-uniformly convex set with respect to $||\cdot||$ with $q \geq 2$. Then the iterates of the Frank-Wolfe algorithm, with short step or exact line search, satisfy*

$$f(x_T) - f(x^*) \leq M/(T + k)^{1/(1-2\theta/q)}, \tag{3.10}$$

with $k \triangleq (2 - 2^\eta)/(2^\eta - 1)$ and $M \triangleq max\{h_0 k^{1/\eta}, 2/((\eta - (1 - \eta)(2^\eta - 1))C)^{1/\eta}\}$, where $\eta \triangleq 1 - 2\theta/q$ and $C \triangleq (\alpha/\mu)^{2/q}/L$. In particular for $q = 2$ and $\theta = 1/2$, we obtain the $\mathcal{O}(1/T^2)$ of [Garber and Hazan, 2015].

Proof. From the proof of Theorem 3.2.2, L-smoothness and the step size decision we have

$$h(x_{t+1}) \leq h(x_t) - \frac{g(x_t)}{2} \cdot \min\Big\{1; \frac{g(x_t)}{L||x_t - v_t||^2}\Big\}.$$

Then using Lemma 3.2.9, we can rewrite

$$\frac{g(x_t)}{||x_t - v_t||^2} = \Big(\frac{g(x_t)^{q/2-1} g(x_t)}{||x_t - v_t||^q}\Big)^{2/q} \geq \Big(\alpha/\mu\Big)^{2/q} g(x_t)^{1-2/q} h_t^{(1-\theta)2/q}.$$

And because $g(x_t) \geq h_t$, we have

$$\frac{g(x_t)}{||x_t - v_t||^2} \geq \Big(\alpha/\mu\Big)^{2/q} h_t^{1-2\theta/q}.$$

We finally end up with the following recursion

$$h(x_{t+1}) \leq h(x_t) \cdot \max\Big\{\frac{1}{2}; 1 - \Big(\alpha/\mu\Big)^{2/q} h_t^{1-2\theta/q}/L\Big\},$$

and we conclude with Lemma 3.A.1. ∎

Overall, Theorem 3.2.2, Theorem 3.2.5 and Theorem 3.2.10 give an (almost) complete picture of all the accelerated convergence regimes one can expect with the vanilla Frank-Wolfe algorithm.

3.3 Online Learning with Linear Oracles and Uniform Convexity

In online convex optimization, the algorithm sequentially decides an action, a point x_t in a set $\mathcal{C}$, and then incurs a (convex smooth) loss $l_t(x_t)$. Algorithms are designed to reduce the cumulative incurred losses over time, $F_t = \frac{1}{t}\sum_{\tau=1}^{t} l_\tau(x_\tau)$. The comparison to the best action in hindsight is then defined as the *regret* of the algorithm, *i.e.* $R_T \triangleq \sum_{t=1}^{T} l_t(x_t) - \min_{x\in\mathcal{C}} \sum_{t=1}^{T} l_t(x)$.

Interesting correspondences have been established between the Frank-Wolfe algorithm and online learning algorithms. For instance, recent works [Abernethy and Wang, 2017, Abernethy et al., 2018] derive new Frank-Wolfe-like algorithms and analyses via two online learning algorithms playing against each other. Furthermore, a series of work proposed projection-free online algorithms inspired by their offline counterpart, *e.g.* Hazan and Kale [2012] design a Frank-Wolfe online algorithm. In following works, Garber and Hazan [2013b,a] propose projection-free algorithms for online and offline optimization with optimal convergence guarantees where the decision sets are polytopes and the loss functions are strongly-convex. In the same setting, Lafond et al. [2015] analyze the online equivalent of the away-step Frank-Wolfe algorithm via a similar analysis to [Lacoste-Julien and Jaggi, 2013, 2015b] in the offline setting. Recently, Hazan and Minasyan [2020] proposed a randomized projection-free algorithm that has a regret of $\mathcal{O}(T^{2/3})$ with high probability improving over the deterministic $\mathcal{O}(T^{3/4})$ of [Hazan and Kale, 2012] and Levy and Krause [2019] designed a projection-free online algorithm over smooth decision sets; dual to uniformly convex sets [Vial, 1983].

Online Linear Optimization and Set Curvature. At a high level, when the constraint set is strongly-convex, the analyses of the simple Follow-The-Leader (FTL) for online linear optimization [Huang et al., 2016b] is analogous to the offline convergence analyses of the Frank-Wolfe algorithm when not assuming strong-convexity of the objective function as in [Polyak, 1966, Demyanov and Rubinov, 1970, Dunn, 1979]. Indeed, by definition, linear functions do not enjoy non-linear lower bounds, *i.e.* uniform convexity-like assumptions.

In the online linear setting, we write the functions $l_t(x) = \langle c_t; x\rangle$ and assume that (c_t) belong to a bounded set $\mathcal{W}$ (smoothness). FTL consists in choosing the action x_t at time t that minimizes the cumulative sum of the previously observed losses, *i.e.* each iteration solves the minimization of a linear function over $\mathcal{C}$

$$x_T \in \operatorname*{argmin}_{x\in\mathcal{C}} \sum_{t=1}^{T-1} l_t(x) = \langle \sum_{t=1}^{T-1} c_t; x\rangle. \tag{3.11}$$

In general, FTL incurs a worst-case regret of $\mathcal{O}(T)$ [Shalev-Shwartz et al., 2012]. For online linear learning, Huang et al. [2016b, 2017] study the conditions under which the strong convexity of the decision set $\mathcal{C}$ leads to improved regret bounds. In particular, when there exists a $c > 0$ such that for all T, $\min_{1\leq t\leq T} ||\frac{1}{t}\sum_{\tau=1}^{t} c_\tau||_* \geq c > 0$, then FTL enjoys the optimal regret bound of $\mathcal{O}(\log(T))$ [Huang et al., 2017]. This result is the counter part of the offline geometrical convergence analyses of the Frank-Wolfe algorithm when $\inf_{x\in\mathcal{C}} ||\nabla f(x)||_* \geq c > 0$ and $\mathcal{C}$ is a strongly convex set [Polyak, 1966, Demyanov and Rubinov, 1970, Dunn, 1979]. In Theorem 3.3.1, we hence further support this analogy between online and offline settings. We show that FTL enjoys continuously interpolated regret bounds between $\mathcal{O}(\log(T))$ and $\mathcal{O}(T)$ for all types of uniform convexity of the decision sets. Again, this covers a much broader spectrum of *curved* sets, and is similar to Theorem 3.2.2 in the Frank-Wolfe setting. A proof is deferred to Appendix 3.C.

Theorem 3.3.1. *Let $\mathcal{C}$ be a compact and (α, q)-uniformly convex set with respect to $||\cdot||$. Assume that $L_T = min_{1\le t\le T}||\frac{1}{t}\sum_{\tau=1}^{t} c_\tau||_* > 0$. Then the regret R_T of FTL (3.11) for online linear optimization satisfies*

$$\begin{cases} R_T \le 2M\Big(\dfrac{2M}{\alpha L_T}\Big)^{1/(q-1)}\Big(\dfrac{q-1}{q-2}\Big)T^{1-1/(q-1)} & \text{when } q > 2 \\ R_T \le \dfrac{4M^2}{\alpha L_T}(1+\log(T)) & \text{when } q = 2, \end{cases} \tag{3.12}$$

where $M = sup_{c\in\mathcal{W}}||c||_*$, with the losses $l_t(x) = \langle c_t; x\rangle$ and (c_t) belong to the bounded set $\mathcal{W}$.

The following is the generalization of [Huang et al., 2017, (6)] when the set is uniformly convex (see Definition 3.1.1). Note that in our version $\mathcal{C}$ can be uniformly convex with respect to any norm. The proof is deferred to Appendix 3.C.

Lemma 3.3.2. *Assume $\mathcal{C} \subset \mathbb{R}^d$ is a (α, q)-uniformly convex set with respect to $||\cdot||$, with $\alpha > 0$ and $q \ge 2$. Consider the non-zero vectors $\phi_1, \phi_2 \in \mathbb{R}^d$ and $v_{\phi_1} \in \operatorname{argmax}_{v\in\mathcal{C}}\langle\phi; v\rangle$ and $v_{\phi_2} \in \operatorname{argmax}_{v\in\mathcal{C}}\langle\phi; v\rangle$. Then*

$$\langle v_{\phi_1} - v_{\phi_2}; \phi_1\rangle \le \Big(\frac{1}{\alpha}\Big)^{1/(q-1)} \frac{||\phi_1 - \phi_2||_*^{1+1/(q-1)}}{(max\{||\phi_1||_*, ||\phi_2||_*\})^{1/(q-1)}}, \tag{3.13}$$

where $||\cdot||_*$ is the dual norm to $||\cdot||$.

Proof of Theorem 3.3.1. The proof follows exactly that of [Huang et al., 2017, Theorem 5]. Write $M = \sup_{c\in\mathcal{F}}||c||$, $F_t(x) = \frac{1}{t}\sum_{\tau=1}^{t}\langle c_t; x\rangle$ and short cut $\nabla F_t \triangleq \frac{1}{t}\sum_{\tau=1}^{t} c_t$ the gradient of the linear function $F_t(x)$. Recall that with FTL, x_t is defined as

$$x_t \in \operatorname{argmin}_{x\in\mathcal{C}}\langle\sum_{\tau=1}^{t-1} c_t; x\rangle.$$

As in [Huang et al., 2017, Theorem 5] we have (for any norm $||\cdot||$)

$$||\nabla F_t - \nabla F_{t-1}|| \le \frac{2M}{t}.$$

Using [Huang et al., 2017, Proposition 2] and Lemma 3.C.1 we get the following upper bound on the regret

$$R_T = \sum_{t=1}^{T} t\langle x_{t+1} - x_t; \nabla F_t\rangle \le \Big(\frac{1}{\alpha}\Big)^{1/(q-1)} \sum_{t=1}^{T} t\frac{||\nabla F_t - \nabla F_{t-1}||_*^{1+1/(q-1)}}{(\max\{||\nabla F_t||_*, ||\nabla F_{t-1}||_*\})^{1/(q-1)}}.$$

Hence, with $L_T = \min_{1\le t\le T}||\nabla F_t||_* > 0$, we have

$$R_T \le 2M\Big(\frac{2M}{\alpha L_T}\Big)^{1/(q-1)} \sum_{t=1}^{T} t^{-1/(q-1)}.$$

Then we have for $q > 2$

$$\sum_{t=1}^{T} t^{-1/(q-1)} = 1 + \sum_{t=2}^{T} t^{-1/(q-1)} \le 1 + \int_{x=1}^{T-1} x^{-1/(q-1)}dx = 1 + \Big[\frac{t^{1-1/(q-1)}}{1-1/(q-1)}\Big]_1^{T-1},$$

so that finally

$$R_T \leq 2M\Big(\frac{2M}{\alpha L_T}\Big)^{1/(q-1)}\Big(\frac{q-1}{q-2}\Big)T^{1-1/(q-1)}.$$

■

With the simple FTL, we obtain non-trivial regret bounds, *i.e.* $o(T)$, whenever the set is uniformly convex, without any curvature assumption on the loss functions (because they are linear). In particular for $q \in [2,3]$, it improves over the general tight regret bound of $\mathcal{O}(\sqrt{T})$ for smooth convex losses and compact convex decision sets [Shalev-Shwartz et al., 2012]. Interestingly, with the same assumption on $\mathcal{C}$, Dekel et al. [2017] obtain for online linear optimization, the same asymptotical regret bounds with a variation of Follow-The-Leader incorporating hints. It is remarkable that the presence of hints or the assumption $\min_{1\leq t\leq T}||\frac{1}{t}\sum_{\tau=1}^{t} c_\tau||_* \geq c > 0$ for all T both lead to the same bounds.

3.4 Examples of Uniformly Convex Objects

The uniform convexity assumptions refine the convex properties of several mathematical objects, such as normed spaces, functions, and sets. In this section, we provide some connection between these various notions of uniform convexity. In Section 3.4.1, we recall that norm balls of uniformly convex spaces are uniformly convex sets, and show set uniform convexity of classic norm balls in Section 3.4.2 and illustrate it with numerical experiments in Section 3.5. In Appendix 3.D.2, we show that the level sets of some uniformly convex functions are uniformly convex sets, extending the strong convexity results of [Garber and Hazan, 2015, Section 5].

3.4.1 Uniformly Convex Spaces

The uniform convexity of norm balls (Definition 3.1.1) is closely related to the uniform convexity of normed spaces [Polyak, 1966, Balashov and Repovs, 2011, Lindenstrauss and Tzafriri, 2013, Weber and Reisig, 2013]. Some works establish sharp uniform convexity results for classical normed spaces such as l_p, L_p or C_p. Most of the practical examples of uniformly convex sets are norm balls and are hence tightly linked with uniformly convex spaces. The property of these sets has many consequences, *e.g.* [Donahue et al., 1997b]. It also relates to concentration inequalities in Banach Spaces [Juditsky and Nemirovski, 2008] and hence implications [Ivanov, 2019] for approximate versions of the Carathéodory theorem [Combettes and Pokutta, 2019].

[Clarkson, 1936, Boas Jr, 1940] define a uniformly convex normed space $(\mathbb{X}, ||\cdot||)$ as a normed space such that, for each $\epsilon > 0$, there is a $\delta > 0$ such that if x and y are unit vectors in $\mathbb{X}$ with $||x-y|| \geq \epsilon$, then $(x+y)/2$ has norm lesser or equal to $1-\delta$. Specific quantification of spaces satisfying this property is obtained via the modulus of convexity, a measure of non-linearity of a norm.

Definition 3.4.1 (Modulus of convexity)**.** *The modulus of convexity of the space* $(\mathbb{X}, ||\cdot||)$ *is defined as*

$$\delta_{\mathbb{X}}(\epsilon) = inf\Big\{1 - \Big|\Big|\frac{x+y}{2}\Big|\Big| \;\Big|\; ||x|| \leq 1\ ,\ ||y|| \leq 1\ ,\ ||x-y|| \geq \epsilon\Big\}. \tag{3.14}$$

A normed space $\mathbb{X}$ is said to be r-uniformly convex in the case $\delta_{\mathbb{X}}(\epsilon) \geq C\epsilon^r$. These specific lower bounds on the modulus of convexity imply that the balls stemming for such spaces are

uniformly convex in the sense of Definition 3.1.1. There exist sharp results for L_p and ℓ_p spaces in [Clarkson, 1936, Hanner, 1956]. Matrix spaces with p-Schatten norm are known as C_p spaces, and sharp results concerning their uniform convexity can be found in [Dixmier, 1953, Tomczak-Jaegermann, 1974, Simon et al., 1979, Ball et al., 1994]. The following gives a link between the set $\gamma_{\mathcal{C}}$ and space $\delta_{\mathbb{X}}$ modulus of convexity, see proof in Appendix 3.D.1.

Lemma 3.4.2. *If a normed space $(\mathbb{X}, ||\cdot||)$ is uniformly convex with modulus of convexity $\delta_{\mathbb{X}}(\cdot)$, then its unit norm ball is $\delta_{\mathbb{X}}(\cdot)$ uniformly convex with respect to $||\cdot||$. Note that if the unit ball $B_{||\cdot||}(1)$ is (α, q)-uniformly convex, then $B_{||\cdot||}(r)$ is $(\alpha/r^{q-1}, q)$-uniformly convex.*

3.4.2 Uniform Convexity of Some Classic Norm Balls

When $p \in]1, 2]$, ℓ_p-balls are strongly convex sets and $((p-1)/2, 2)$-uniformly convex with respect to $||\cdot||_p$, see for instance [Hanner, 1956, Theorem 2] or [Garber and Hazan, 2015, Lemma 4]. When $p > 2$, the ℓ_p-balls are $(1/p, p)$-uniformly convex with respect to $||\cdot||_p$ [Hanner, 1956, Theorem 2]. Uniform convexity also extends the strong convexity of group $\ell_{s,p}$-norms (with $1 < p, s \leq 2$) [Garber and Hazan, 2015, §5.3. and 5.4.] to the general case $p, s > 1$.

[Dixmier, 1953, Tomczak-Jaegermann, 1974, Simon et al., 1979, Ball et al., 1994] focus of the uniform convexity of the $(C_p, ||\cdot||_{S(p)})$ spaces, *i.e.* spaces of matrix where the norm is the ℓ_p-norm of a matrix singular values . Their unit balls are hence the p-Schatten balls. For $p \in]1, 2]$, p-Schatten balls are $((p-1)/2, 2)$-uniformly convex with respect to $||\cdot||_{S(p)}$, see [Garber and Hazan, 2015, Lemma 6] and the sharp results of [Ball et al., 1994]. For the case $p > 2$, [Dixmier, 1953] showed that the p-Schatten balls are $(1/p, p)$-uniformly convex with respect to $||\cdot||_{S(p)}$, see also [Ball et al., 1994, §III].

3.5 Numerical Illustration

Uniform convexity is a global assumption. Hence, in Theorem 3.2.2, we obtain sublinear convergence that do not depend on the specific location of the solution $x^* \in \partial\mathcal{C}$. However, some regions of $\mathcal{C}$ might be relatively more curved than others and hence exhibit faster convergence rates. This effect is quantified in Theorem 3.2.5 when a local scaling inequality holds.

In Figure 3.1, in the case of the ℓ_p-balls with $p > 2$, we vary the approximate location of the optimum x^* in the boundary of the ℓ_p-balls.

Subfigures (3.1a), (3.1b), and (3.1c) are associated to an optimization problem where the solution x^* of (OPT) is near the intersection of the ℓ_p-balls and the half-line generated by $\sum_{i=1}^{d} e_i$ (where the (e_i) is the canonical basis), *i.e.* in *curved* regions of the boundaries of the ℓ_p-balls.

Subfigures (3.1d), (3.1e), and (3.1f) corresponds to the same optimization problem where the solution x^* to (OPT) is close to the intersection between the half-line generated by e_1 and the boundary of the ℓ_p-balls, *i.e.* in *flat* regions of the boundaries of the ℓ_p-balls.

We observe that when the optimum is at a *curved* location, the convergence is quickly linear for p sufficiently close to 2 and appropriate line-search (see Subfigures (3.1b) and (3.1c)). However, when the optimum is near the *flat* location, we indeed observe sublinear convergence rates (see Subfigures (3.1e) and (3.1f)). It still becomes linear for $p = 2.1$ with exact line-search in Subfigure (3.1f).

Also, Theorem 3.2.2 gives accelerated rates when using the Frank-Wolfe algorithm with exact line-search or short step. In Subfigures (3.1a) and (3.1d), we show examples of the

convergence of the Frank-Wolfe algorithm when using deterministic line-search. The rates are indeed sublinear in $\mathcal{O}(1/T)$. In other words, deterministic line-search generally do not lead to accelerated convergence rates when the sets are uniformly convex.

3.6 Conclusion

Our results fill the gap between known convergence rates for the Frank Wolfe algorithm. Qualitatively, they also mean that it is the *curvature* of the constraint set that accelerates the convergence of the Frank-Wolfe algorithm, not just strong-convexity. This emphasis on curvature echoes works in other settings [Huang et al., 2016b]. For the sake of theory, the results could be immediately refined by measuring the *local* curvature of convex bodies with more sophisticated tools than uniform convexity [Schneider, 2015].

From a more practical perspective, uniform convexity encompasses ubiquitous structures of constraint sets appearing in machine learning and signal processing. In applications where the (e.g. regularization) constraints are likely to be active, the assumption that $\inf_{x\in\mathcal{C}}||\nabla f(x)||_* > 0$ is not restrictive and the value of c quantifies the relevance of the constraints.

Overall our results go back to the basics. They show that the Frank-Wolfe mechanism, *i.e.* minimizing the linear approximation of the function and doing the right convex update, leads to accelerated convergence rates for a large variety of *curved* sets, the uniformly convex sets, in a fully-adaptive fashion.

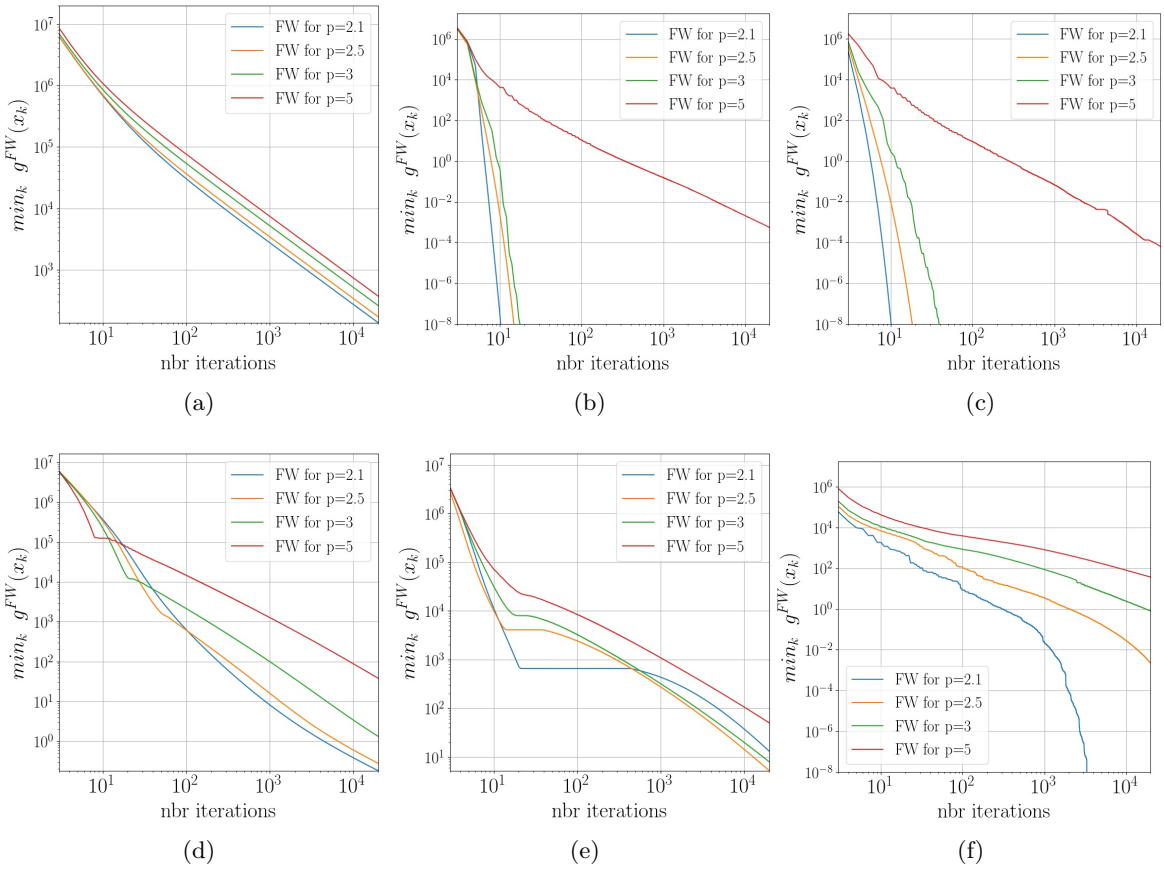


Figure 3.1: Solving (OPT) with the Frank-Wolfe algorithm where f is a quadratic with condition number 100 and the constraint sets are various ℓ_p-balls of radius 5. We vary p so that all balls are uniformly convex but not strongly-convex. We vary the position of the solution to (OPT) with respect to the boundaries of the constraints sets. On the first row, we choose the constrained optimum close to the intersection of the set boundary and the line generated by $\sum_i e_i$ (where the e_i form the canonical basis), where ℓ_p-balls are typically more *curved*. On the second row, we choose the constrained optimum near the intersection between the set boundary and the line generate by e_1, a region where the ℓ_p-balls are *flat*. On a line, each plot exhibits the behavior of the Frank-Wolfe algorithm iterates with different step size strategy: deterministic line-search (i.e. $1/(k+1)$), short step and exact line-search. To avoid the oscillating behavior of Frank-Wolfe gap, the y-axis represents $\min_{k=1,\dots,T} g(x_k)$ where $g(\cdot)$ is the Frank-Wolfe gap and T the number of iterations.

Appendices

3.A Recursive Lemma

The proofs of Theorems 3.2.2, 3.2.5, and 3.2.10 involve finding explicit bounds for sequences (h_t) satisfying recursive inequalities of the form,

$$h_{t+1} \leq h_t \cdot \max\{1/2, 1 - Ch_t^\eta\}. \tag{3.15}$$

with $\eta < 1$. An explicit solution with $\eta = 1/2$ is given in [Garber and Hazan, 2015] and corresponds to $h_t = \mathcal{O}(1/T^2)$, while for $\eta = 1$ we recover the classical sublinear Frank-Wolfe regime of $\mathcal{O}(1/T)$. For a $\eta \in]0, 1]$, we have $\mathcal{O}(1/T^{1/\eta})$ (see for instance [Temlyakov, 2011] or [Nguyen and Petrova, 2017, Lemma 4.2.]), which can be guessed via $h(t) = (C\eta)^{1/\eta}t^{-1/\eta}$ the solution of the differential equation $h'(t) = -Ch(t)^{\eta+1}$ for $t > 0$. A quantitative statement is, for instance, given in [Xu and Yang, 2018, proof of Theorem 1.] that we reproduce here.

Lemma 3.A.1 (Recurrence and sub-linear rates)**.** *Consider a sequence $(h_t)_{t\in\mathbb{N}}$ of non-negative numbers satisfying (3.15) with $0 < \eta \leq 1$, then $h_T = \mathcal{O}(1/T^{1/\eta})$. More precisely for all $t \geq 0$,*

$$h_t \leq \frac{M}{(t+k)^{1/\eta}}$$

with $k \triangleq (2 - 2^\eta)/(2^\eta - 1)$ and $M \triangleq max\{h_0 k^{1/\eta}, 2/((\eta - (1-\eta)(2^\eta - 1))C)^{1/\eta}\}$.

3.B Beyond Local Uniform Convexity

Here we show that additional convexity properties on the gauge function of $\mathcal{C}$ imply local scaling inequalities on $\mathcal{C}$. Note that for ease, we assume that the gauge function is differential at x^* which is not necessarily the case case when the set $\mathcal{C}$ is uniformly convex.

Lemma 3.B.1. *Consider a compact convex set $\mathcal{C}$ with 0 in its interior. Assume the gauge function of $\mathcal{C}$ is differentiable and normal cone at the boundary are half-lines. Assume (μ, r)-uniformly convex at x^* a solution of (OPT) (where f is a convex L-smooth function and $inf_{x\in\mathcal{C}}||x||_{\mathcal{C}} > 0$), then we have the following scaling inequality for all $x \in \mathcal{C}$*

$$\langle -\nabla f(x^*); x - x^* \rangle \geq \frac{\mu}{||g||}||\nabla f(x^*)||||x - x^*||^q,$$

where $g \in N_{\mathcal{C}}(x^*)$ and $\langle g; x^* \rangle = 1$.

Proof. We have $x^* \in \partial\mathcal{C}$. Write $g = \nabla||x||_{\mathcal{C}}$. Then by (μ, r)-uniformly convex of the gauge function we have

$$||x||_{\mathcal{C}} \geq \underbrace{||x^*||_{\mathcal{C}}}_{=1} + \langle g; x - x^* \rangle + \mu||x - x^*||^q.$$

Hence we have

$$\langle -g; x - x^* \rangle \geq \underbrace{1 - ||x||_{\mathcal{C}}}_{\geq 0} + \mu||x - x^*||^q \geq \mu||x - x^*||^q.$$

When it is differentiable, [Schneider, 2014, (1.39)] show that g satisfies $g \in N_{\mathcal{C}}(x^*)$ and $\langle g; x^* \rangle = 1$. Here, the normal cone is a half-line and $-\nabla f(x^*) \in N_{\mathcal{C}}(x^*)$. In particular then $-\nabla f(x^*) = \frac{||\nabla f(x^*)||}{||g||} g$. Finally

$$\langle -\nabla f(x^*); x - x^* \rangle \geq \frac{\mu}{||g||} ||x - x^*||^q ||\nabla f(x^*)||.$$

■

3.C Proofs in Online Optimization

The following is the generalization of [Huang et al., 2017, (6)] when the set is uniformly convex. Note that in our version $\mathcal{C}$ can be uniformly convex with respect to any norm.

Lemma 3.C.1. *Assume $\mathcal{C} \subset \mathbb{R}^d$ is a (α, q)-uniformly convex set with respect to $||\cdot||$, with $\alpha > 0$ and $q \geq 2$. Consider the non-zero vectors $\phi_1, \phi_2 \in \mathbb{R}^d$ and $v_{\phi_1} \in \operatorname{argmax}_{v \in \mathcal{C}} \langle \phi; v \rangle$ and $v_{\phi_2} \in \operatorname{argmax}_{v \in \mathcal{C}} \langle \phi; v \rangle$. Then*

$$\langle v_{\phi_1} - v_{\phi_2}; \phi_1 \rangle \leq \left(\frac{1}{\alpha}\right)^{1/(q-1)} \frac{||\phi_1 - \phi_2||_*^{1+1/(q-1)}}{(max\{||\phi_1||_*, ||\phi_2||_*\})^{1/(q-1)}}, \tag{3.16}$$

where $||\cdot||_*$ is the dual norm to $||\cdot||$.

Proof. By definition of uniform convexity, for any z of unit norm, $v_\gamma(z) \in \mathcal{C}$ where

$$v_\gamma(z) = \gamma v_{\phi_1} + (1-\gamma) v_{\phi_2} + \gamma(1-\gamma)\alpha ||v_{\phi_1} - v_{\phi_2}||^q z.$$

By optimality of v_{ϕ_1} and v_{ϕ_2}, we have $\langle v_\gamma(z); \phi_1 \rangle \leq \langle v_1; \phi_1 \rangle$ and $\langle v_\gamma(z); \phi_2 \rangle \leq \langle v_2; \phi_2 \rangle$, so that

$$\langle v_\gamma(z); \gamma\phi_1 + (1-\gamma)\phi_2 \rangle \leq \gamma \langle v_1; \phi_1 \rangle + (1-\gamma)\langle v_2; \phi_2 \rangle.$$

Write $\phi_\gamma = \gamma\phi_1 + (1-\gamma)\phi_2$. Then, when developing the left hand side, we get

$$\gamma(1-\gamma)\alpha ||v_{\phi_1} - v_{\phi_2}||^q \langle z; \phi_\gamma \rangle \leq \gamma(1-\gamma)\langle v_{\phi_1} - v_{\phi_2}; \phi_1 - \phi_2 \rangle$$

Choosing the best z of unit norm we get

$$\alpha ||v_{\phi_1} - v_{\phi_2}||^q ||\phi_\gamma||_* \leq \langle v_{\phi_1} - v_{\phi_2}; \phi_1 - \phi_2 \rangle$$

and for $\gamma = 0$ and $\gamma = 1$ and via generalized Cauchy-Schwartz we get

$$\alpha ||v_{\phi_1} - v_{\phi_2}||^q \cdot \max\{||\phi_1||_*, ||\phi_2||_*\} \leq ||v_{\phi_1} - v_{\phi_2}|| \cdot ||\phi_1 - \phi_2||_*.$$

Then,

$$\langle v_{\phi_1} - v_{\phi_2}; \phi_1 \rangle \leq ||v_{\phi_1} - v_{\phi_2}|| \cdot ||\phi_1 - \phi_2||_* \leq \left(\frac{1}{\alpha}\right)^{1/(q-1)} \frac{||\phi_1 - \phi_2||_*^{1+1/(q-1)}}{(\max\{||\phi_1||_*, ||\phi_2||_*\})^{1/(q-1)}},$$

and we finally obtain (3.16). ■

3.D Uniformly Convex Objects

3.D.1 Uniformly Convex Spaces

Proof of Lemma 3.4.2. Assume $(\mathbb{X}, ||\cdot||)$ is uniformly convex with modulus of convexity $\delta(\cdot)$. Then for any $(x, y, z) \in B_{||\cdot||}(1)$, we have by definition $1 - \frac{||x+y||}{2} \geq \delta(||x-y||)$ and then

$$\left|\left|\frac{x+y}{2} + \delta(||x-y||)z\right|\right| \leq \left|\left|\frac{x+y}{2}\right|\right| + \delta(||x-y||) \leq 1 \ .$$

Hence, $\frac{x+y}{2} + \delta(||x-y||)z \in \mathcal{C}$. Without loss of generality, consider $\eta \in]0; 1/2]$. We need to show that $\eta x + (1-\eta)y + \delta(||x-y||)z \in \mathcal{C}$ for any z with norm lesser than 1. First, note that $\eta x + (1-\eta)y = (1-2\eta)y + (2\eta)(x+y)/2$. Note also that because $1 - 2\eta \in [0, 1]$, we have for any z of norm lesser than 1

$$(1-2\eta)x + (2\eta)\left[(x+y)/2 + \delta(||x-y||)z\right] \in \mathcal{C}.$$

Hence, for any z of norm lesser than 1, we have

$$\eta x + (1-\eta)y + 2\eta\delta(||x-y||)z \in \mathcal{C}.$$

Or equivalently

$$\eta x + (1-\eta)y + (1-\eta)\eta\delta(||x-y||)\frac{2\eta}{(1-\eta)\eta}z \in \mathcal{C}.$$

Because $\frac{2\eta}{(1-\eta)\eta} \geq 1$, it follows that for any z of norm lesser than 1 we have

$$\eta x + (1-\eta)y + (1-\eta)\eta\delta(||x-y||)z \in \mathcal{C},$$

which conclude on the uniform convexity of the norm ball. ■

3.D.2 Uniformly Convex Functions

Uniform convexity is also a property of convex functions and defined as follows.

Definition 3.D.1. *A differentiable function f is (μ, r)-uniformly convex on a convex set $\mathcal{C}$ if there exists $r \geq 2$ and $\mu > 0$ such that for all $(x, y) \in \mathcal{C}$*

$$f(y) \geq f(x) + \langle \nabla f(x); y - x\rangle + \frac{\mu}{2}||x-y||_2^r \ .$$

We now state the equivalent of [Journée et al., 2010, Theorem 12] for the level sets of uniformly convex functions. This was already used in [Garber and Hazan, 2015] in the case of strongly-convex sets.

Lemma 3.D.2. *Let $f : \mathbb{R}^d \to \mathbb{R}^+$ be a non-negative, L-smooth and (μ, r)-uniformly convex function on $\mathbb{R}^d$, with $r \geq 2$. Then for any $w > 0$, the set*

$$\mathcal{L}_w = \left\{x \mid f(x) \leq w\right\} \ ,$$

is (α, r)-uniformly convex with $\alpha = \frac{\mu}{\sqrt{2wL}}$.

Proof. The proof follows exactly that of [Journée et al., 2010, Theorem 12], replacing $||x-y||^2$ with $||x-y||^r$. We state it for the sake of completeness. Consider $w_0 > 0$, $(x, y) \in \mathcal{L}_w$ and $\gamma \in [0, 1]$. We denote $z = \gamma x + (1-\gamma)y$. For $u \in \mathbb{R}^d$, by L-smoothness applied at z and at x^* (the unconstrained optimum of f), we have

$$\begin{aligned} f(z+u) &\leq f(z) + \langle \nabla f(z); u \rangle + \frac{L}{2}||u||_2^2 \\ &\leq f(z) + ||\nabla f(z)|| \cdot ||u|| + \frac{L}{2}||u||_2^2 \\ &\leq f(z) + \sqrt{2Lf(z)}||u|| + \frac{L}{2}||u||_2^2 = \left(\sqrt{f(z)} + \sqrt{\frac{L}{2}}||u||\right)^2 . \end{aligned}$$

Note that uniform convexity of f implies that

$$f(z) \leq \gamma f(x) + (1-\gamma)f(y) - \frac{\mu}{2}\gamma(1-\gamma)||x-y||^r$$

In particular then, because $x, y \in \mathcal{L}_w$, we have $f(z) \leq w - \frac{\mu}{2}\gamma(1-\gamma)||x-y||^r$ so that

$$f(z+u) \leq \left(\sqrt{w - \frac{\mu}{2}\gamma(1-\gamma)||x-y||^r} + \sqrt{\frac{L}{2}}||u||\right)^2 \tag{3.17}$$

Leveraging on the concavity of the square-root, we get

$$f(z+u) \leq \left(\sqrt{w} - \frac{\mu}{4\sqrt{w}}\gamma(1-\gamma)||x-y||^r + \sqrt{\frac{L}{2}}||u||\right)^2 . \tag{3.18}$$

Hence for any u such that

$$||u|| = \frac{\mu}{2\sqrt{2wL}}\gamma(1-\gamma)||x-y||^r ,$$

we have $z + u \in \mathcal{L}_w$. Hence $\mathcal{L}_w$ is a $(\frac{\mu}{2\sqrt{2wL}}, r)$-uniformly convex set. ∎

Lemma 3.D.2 restrictively requires smoothness of the uniformly convex function f. Hence we provide the analogous of [Garber and Hazan, 2015, Lemma 3].

Lemma 3.D.3. *Consider a finite dimensional normed vector space $(\mathbb{X}, ||\cdot||)$. Assume $f(x) = ||x||^2$ is (μ, s)-uniformly convex function (with $r \geq 2$) with respect to $||\cdot||$. Then the norm balls $B_{||\cdot||}(r) = \left\{x \in \mathbb{X} \mid ||x|| \leq r\right\}$ are $(\frac{\mu}{2r}, s)$-uniformly convex.*

Proof. The proof follows exactly that of [Garber and Hazan, 2015, Lemma 3] which itself follows that of [Journée et al., 2010, Theorem 12], where operations involving L-smoothness are replaced by an application of the triangular inequality.

Let's consider $s \geq 2$, $(x, y) \in B_{||\cdot||}(r)$ and $\gamma \in [0, 1]$. We denote $z = \gamma x + (1-\gamma)y$. For $u \in \mathbb{X}$, applying successively triangular inequality and (μ, s)-uniform convexity of $f(x) = ||x||^2$, we get

$$\begin{aligned} f(z+u) = ||z+u||^2 &\leq \left(\sqrt{f(z)} + ||u||\right)^2 \\ &\leq \left(\sqrt{r^2 - \frac{\mu}{2}\gamma(1-\gamma)||x-y||^s} + ||u||\right)^2 . \end{aligned}$$

We then use concavity of the square root as before to get

$$||z+u||^2 \leq \left(r - \frac{\mu}{4r}\gamma(1-\gamma)||x-y||^s + ||u||\right)^2 .$$

In particuler, for $u \in \mathbb{X}$ such that $||u|| = \frac{\mu}{4r}\gamma(1-\gamma)||x-y||^s$, we have $z+u \in B_{||\cdot||}(r)$. Hence $B_{||\cdot||}(r)$ is $(\frac{\mu}{2r}, s)$- uniformly convex with respect to $||\cdot||$. ■

These previous lemmas hence allow to translate functional uniformly convex results into results for classic balls norms. For instance, [Shalev-Shwartz, 2007, Lemma 17] showed that for $p \in]1,2]$ $f(x) = 1/2||x||_p^2$ was $(p-1)$-uniformly convex with respect to $||\cdot||_p$.

Chapter 4

Subsampling Frank-Wolfe

In this chapter we analyze two novel randomized variants of Frank-Wolfe (FW) or conditional gradient algorithms. While classical FW algorithms require solving a linear minimization problem over the domain at each iteration, the proposed method only requires to solve a linear minimization problem over a small *subset* of the original domain. The first algorithm that we propose is a randomized variant of the original FW algorithm and achieves a $\mathcal{O}(1/T)$ sublinear convergence rate as in the deterministic counterpart on compact convex domains. The second algorithm is a randomized variant of the Away-step FW algorithm, and again as its deterministic counterpart, reaches linear convergence rate on polytopes making it the first provably convergent randomized variant of Away-step FW. In both cases, while subsampling reduces the convergence rate by a constant factor, the cost of the linear minimization step can be a fraction of the deterministic versions, especially when the data is streamed. We illustrate computational gains on regression problems, involving both ℓ_1 and latent group lasso penalties.

Contents

4.1 Introduction

As in previous chapters, the Frank-Wolfe (FW) or conditional gradient algorithm [Frank and Wolfe, 1956] is applied to solve optimization problems of the form

$$\underset{x \in \mathcal{C}}{\text{minimize}} \; f(x) \;, \text{ with } \mathcal{C} = \text{conv}(\mathcal{A}) \;, \tag{OPT}$$

where $\mathcal{A}$ is a (possibly infinite) set of vectors which we call *atoms*, and $\text{conv}(\mathcal{A})$ is its convex hull, see Section 1.1.4. Again, the FW algorithms have seen an impressive revival in recent years, due to their low memory requirements and projection-free iterations, which make them particularly appropriate to solve large scale convex problems, for instance convex relaxations of problems written over combinatorial polytopes [Zaslavskiy et al., 2009, Joulin et al., 2014, Vogelstein et al., 2015].

Despite these attractive properties, for problems with a large number of variables or with a very large atomic set (or both), computing the full gradient and LMO at each iteration may become prohibitive. Designing variants of the FW algorithm which alleviate this computational burden would have a significant practical impact on performance.

One recent direction to achieve this is to replace the LMO with a *randomized* linear oracle in which the linear minimization is performed only over a random sample of the original atomic domain. This approach has proven to be highly successful on specific problems such as structured SVMs [Lacoste-Julien et al., 2013] and constrained discriminative clustering [Miech et al., 2017, Peyre et al., 2017, Miech et al., 2018]. However, little is known in the general case. Is it possible to design a FW variant with a randomized oracle that achieves the same convergence rate (up to a constant factor) as the non-randomized variant? Can this be extended to linearly-convergent FW algorithms [Lacoste-Julien and Jaggi, 2013, 2015b, Garber and Hazan, 2015, Pena and Rodriguez, 2018]? In this chapter, we give a positive answer to both questions and explore the trade-offs between subsampling and convergence rate.

Outline and main contribution. The main contribution of this chapter is to develop and analyze two algorithms that share the projection-free iterations of FW, but in which the LMO is computed only over a random subset of the original domain. In many cases, this results in significant gains in computing the LMO which can also speed up the overall FW algorithm. In practice, the algorithm will run a larger number of cheaper iterations, which is typically more efficient for huge data sets (e.g. in a streaming model where the data does not fit in core memory and can only be accessed by chunks). The paper is structured as follows

- §4.2 describes the "Randomized FW" algorithm, proving a sublinear convergence rate.
- §4.3 describes "Randomized Away FW" algorithm, a variant which enjoys a linear convergence rate on polytopes. To the best of our knowledge this is the first provably convergent randomized version of the Away-steps FW algorithm.
- Finally, in §4.4 we discuss implementation aspects of the proposed algorithms and study their performance on lasso and latent group lasso problems.

Note that with the proven sub-linear rate of convergence for Randomized FW (RFW), the cost of the LMO is reduced by the subsampling rate, but this is compensated by the fact that the number of iterations required by RFW to reach same convergence guarantee

as FW is itself multiplied by the sampling rate. However, the linear convergence rate in Randomized AFW does not theoretically show a computational advantage since the number of iterations is multiplied by the squared sampling rate, in our highly conservative bounds at least. Nevertheless, our numerical experiments show that randomized versions are often numerically superior to their deterministic counterparts.

Related work. Several references have focused on reducing the cost of computing the linear minimization oracle. The analysis of [Jaggi, 2013] allows for an error term in the LMO, and so a randomized linear oracle could in principle be analyzed under this framework. However, this is not entirely satisfactory as it requires the approximation error to decrease towards zero as the algorithm progresses. In our algorithm, the subsampling approximation error does not need to decrease.

Lacoste-Julien et al. [2013] studied a randomized FW variant named block-coordinate FW in which at each step the LMO is computed only over a subset (block) of variables. In this case, the approximation error does not need to decrease to zero, but the method can only be applied to a restricted class of problems: those with a block-separable domain, leaving out significant cases such as ℓ_1–constrained minimization for instance. Because of the block separability, a more aggressive step-size strategy can be used in this case, resulting overall in a different algorithm.

Finally, Frandi et al. [2014] proposed a FW variant which is a particular case of our Algorithm 10 for the Lasso problem, analyzed in Frandi et al. [2016]. Our analysis here brings three critical improvements to this last result. First, it is provably convergent for arbitrary atomic domains, not just the ℓ_1 ball. Second, it allows a choice of step size that does not require exact line-search (Variant 2), which is typically only feasible for quadratic loss functions. Third, we extend our analysis to linearly-convergent FW variants such as the Away-step FW.

A different technique to alleviate the cost of the linear oracle was recently proposed by Braun et al. [2017b]. In that work, the authors propose a FW variant that replaces the LMO by a "weak" separation oracle. They showed significant speedups in wall-clock performance on practical problems. This approach was combined with gradient sliding in Lan et al. [2017], a technique [Lan and Zhou, 2016] that allows skipping the computation of gradients from time to time. However, for problems such as Lasso or latent group lasso, a randomized LMO avoids all full gradient computations, while the lazy weak separation oracle still requires it. Combining these various techniques is an interesting open question.

Proximal coordinate-descent methods [Richtárik and Takáč, 2014] (not based on FW) have also been used to solve problems with a huge number of variables. They are particularly effective when associated with variable screening rules such as [Tibshirani et al., 2012, Fercoq et al., 2015]. However, for constrained problems, they require evaluating a projection operator, which on some sets such as the latent group lasso ball can be much more expensive than the LMO. Furthermore, these methods require that the projection operator is block-separable, while our method does not.

Notation. We denote sets in calligraphic letter (i.e., $\mathcal{A}$). We use $\text{clip}_{[0,1]}(s) = \max\{0, \min\{1, s\}\}$. Probability is written $\mathcal{P}$ and cardinality of a set $\mathcal{A}$ is denoted $|\mathcal{A}|$. For x^* a solution of (OPT), we write $h(x) = f(x) - f(x^*)$ the primal gap. FW variants with randomness in the LMO are called *randomized* and we reserve the name *stochastic* for FW variants that replace the gradient with a stochastic approximation, as in [Hazan and Luo, 2016].

4.2 Randomized Frank-Wolfe

In this section we present our first contribution, the Randomized Frank-Wolfe (RFW) algorithm. The method is detailed in Algorithm 10. Compared to the standard FW algorithm, it has the following two distinct features.

First, the LMO is computed over a random subset $\mathcal{A}_t \subseteq \mathcal{A}$ of the original atomic set in which each atom is equally likely to appear, i.e., in which $\mathcal{P}(v \in \mathcal{A}_t) = \eta$ for all $v \in \mathcal{A}$ (Line 3). For discrete sets this can be implemented simply by drawing uniformly at random a fixed number of elements at each iteration. The sampling parameter η controls the fraction of the domain that is considered by the LMO at each iteration. If $\eta = 1$, the LMO considers the full domain at each iteration and the algorithm defaults to the classical FW algorithm. However, for $\eta < 1$, the LMO only needs to consider a fraction of the atoms in the original dataset and can be faster than the FW LMO.

Second, unlike in the FW algorithm, the atom chosen by the LMO is not necessarily a descent direction and so it is no longer possible to use the "oblivious" (i.e., independent on the result of the LMO) $2/(2+t)$ step-size commonly used in the FW algorithm. We provide two possible choices for this step-size: the first variant (Line 5) chooses the step-size by exact line search and requires to solve a 1-dimensional convex optimization problem. This approach is efficient when this sub-problem has a closed form solution, as it happens for example in the case of quadratic loss functions. The second variant does not need to solve this sub-problem, but in exchange requires to have an estimate of the curvature constant C_f (defined in next subsection). Note that in absence of an estimate of this quantity, one can use the bound $C_f \leq \operatorname{diam}(\mathcal{C})^2 L$, where L is the Lipschitz constant of ∇f and $\operatorname{diam}(\mathcal{C})$ is the diameter of the domain in euclidean norm.

Gradient coordinate subsampling. We note that the gradient of f only enters Algorithm 10 through the computation of the randomized LMO, and so only the dot product between the gradient and the subsampled atomic set are truly necessary. In some cases the elements of the atomic set have a specific structure that makes computing dot products particularly effective. For example, when the atomic elements are sparse, only the coordinates of the gradient that are in the support of the atomic set need to be evaluated. As a result, for sparse atomic sets such as the ℓ_1 ball, the group lasso ball (also known as ℓ_1/ℓ_2 ball), or even the latent group lasso [Obozinski et al., 2011] ball, only a few coordinates of the gradient need to be evaluated at each iteration. The number of exact gradients that need to be evaluated will depend on both the sparsity of this atomic set and the subsampling rate. For example, in the case of the ℓ_1 ball, the extreme atoms have a single nonzero coefficient, and so RFW only needs to compute on average $d\eta$ gradient coefficients at each iteration, where d denotes the ambient dimension.

Stopping criterion. A side-effect of subsampling the linear oracle is that $\langle -\nabla f(x_t); s_t - x_t \rangle$, where s_t is the atom selected by the randomized linear oracle is no longer an upper bound on $f(x_t) - f(x^*)$. This property is a feature of FW algorithms that cannot be retrieved in our variant. As a replacement, the stopping criteria that we propose is to compute a full LMO every $k\lfloor \frac{1}{\eta} \rfloor$ iterations, with $k \in \mathbb{N}^*$ ($k = 2$ is a good default value).

Algorithm 10 Randomized Frank-Wolfe algorithm

Input: $x_0 \in \mathcal{C}$, sampling ratio $0 < \eta \leq 1$.

1:
2: **for** $t = 0, 1 \ldots, T$ **do**
3: Choose $\mathcal{A}_t$ such that $\mathcal{P}(v \in \mathcal{A}_t) = \eta$ for all $v \in \mathcal{A}$
4: Compute $s_t = \mathrm{LMO}(\nabla f(x_t), \mathcal{A}_t)$ ▷ subsampled LMO
5: Variant 1: $\gamma_t = \mathrm{argmax}_{\gamma \in [0,1]} f((1-\gamma)x_t + \gamma s_t)$ ▷ exact line-search
6: Variant 2: $\gamma_t = \mathrm{clip}_{[0,1]}(\langle -\nabla f(x_t), s_t - x_t \rangle / C_f)$ ▷ short-step size
7: $x_{t+1} = (1-\gamma_t)x_t + \gamma_t s_t$
8: **end for**

4.2.1 Analysis

In this subsection we prove an $\mathcal{O}(1/t)$ convergence rate for the RFW algorithm. As is often the case for FW-related algorithms, our convergence result will be stated in terms of the *curvature constant* C_f, which is defined as follows for a convex and differentiable function f and a convex and compact domain $\mathcal{C}$:

$$C_f \triangleq \sup_{\substack{x,s \in \mathcal{C}, \gamma \in [0,1] \\ y = x + \gamma(s-x)}} \frac{2}{\gamma^2}(f(y) - f(x) - \langle \nabla f(x), y - x \rangle).$$

It is worth mentioning that a bounded curvature constant C_f corresponds to a Lipschitz assumption on the gradient of f [Jaggi, 2013].

Theorem 4.2.1. *Let f be a function with bounded smoothness constant C_f and subsampling parameter $\eta \in (0, 1]$. Then Algorithm 10 (in both variants) converges towards a solution of* (OPT)*. Furthermore, the following inequality is satisfied:*

$$\mathbb{E}[h(x_T)] \leq \frac{2(C_f + f(x_0) - f(x^*))}{\eta T + 2} \ . \tag{4.1}$$

Proof. See 4.A. ■

The rate obtained in the previous theorem is similar to known bounds for FW. For example, [Jaggi, 2013, Theorem 1] established for FW a bound of the form

$$h(x_T) \leq \frac{2C_f}{T+2} \ . \tag{4.2}$$

This is similar to the rate of Theorem 4.2.1, except for the factor η in the denominator. Hence, if our updates are η times as costly as the full FW update (as is the case e.g. for the ℓ_1 ball), then the theoretical convergence rate is the same. This bound is likely tight, as in the worst case one will need to sample the whole atomic set to decrease the objective if there is only one descent direction. This is however a very pessimistic scenario, and in practice good descent directions can often be found without sampling the whole atomic set. As we will see in the experimental section, despite these conservative bounds, the algorithm often exhibits large computational gains with respect to the deterministic algorithm.

4.3 Randomized Away-steps Frank-Wolfe

A popular variant of the FW algorithm is the Away-steps FW variant of Guélat and Marcotte [1986]. This algorithm 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 have much better convergence properties, such as linear (i.e. exponential) convergence rates for generally-strongly convex objectives [Garber and Hazan, 2013a, Beck and Tetruashvili, 2013, Lacoste-Julien and Jaggi, 2015b].

In this section we describe the first provably convergent randomized version of the Away-steps FW, which we name *Randomized Away-steps FW* (RAFW). We will assume throughout this section that the domain is a polytope, i.e. that $\mathcal{C} = \text{conv}(\mathcal{A})$, where $\mathcal{A}$ is a finite set of atoms. We will make use of the following notation.

- *Active set.* We denote by $\mathcal{S}_t$ the active set of the current iterate, i.e. x_t decomposes as $x_t = \sum_{v \in \mathcal{S}_t} \alpha_v^{(t)} v$, where $\alpha_v^{(t)} > 0$ are positive weights that are iteratively updated.
- *Subsampling parameter.* The method depends on a subsampling parameter p. It controls the amount of computation per iteration of the LMO. In this case, the atomic set is finite and p denotes an integer $1 \leq p \leq |\mathcal{A}|$. This sampling rate is approximately $\lfloor \eta |\mathcal{A}| \rfloor$ in the RFW formulation of §4.2.

The method is described in Algorithm 11 and, as in the Away-steps FW, requires computing two linear minimization oracles at each iteration. Unlike the deterministic version, the first oracle is computed on the subsampled set $\mathcal{S}_t \cup \mathcal{A}_t$ (Line 4), where $\mathcal{A}_t$ is a subset of size $\min\{p, |\mathcal{A} \setminus \mathcal{S}_t|\}$, sampled uniformly at random from $\mathcal{A} \setminus \mathcal{S}_t$. The second LMO (Line 6) is computed on the active set, which is also typically much smaller than the atomic domain.

As a result of both oracle calls, we obtain two potential descent directions, the RFW direction d_t^{FW} and the Away direction d_t^{A}. The chosen direction is the one that correlates the most with the negative gradient, and a maximum step size is chosen to guarantee that the iterates remain feasible (Lines 8–11).

Updating the support. Line 14 requires updating the support and the associated α coefficients. For a FW step we have $\mathcal{S}_{t+1} = \{s_t\}$ if $\gamma_t = 1$ and otherwise $\mathcal{S}_{t+1} = \mathcal{S}_t \cup \{s_t\}$. The corresponding update of the weights is $\alpha_v^{(t+1)} = (1 - \gamma_t)\alpha_v^{(t)}$ when $v \in \mathcal{S}_t \setminus \{s_t\}$ and $\alpha_{s_t}^{(t+1)} = (1 - \gamma_t)\alpha_{s_t}^{(t)} + \gamma_t$ otherwise.

For an away step we instead have the following update rule. When $\gamma_t = \gamma_{\max}$ (which is called a *drop step*), then $\mathcal{S}_{t+1} = \mathcal{S}_t \setminus \{v_t\}$. Combined with $\gamma_{\max} < 1$ (or equivalently $\alpha_{v_t} \leq \frac{1}{2}$) we call them *bad drop step*, as it corresponds to a situation in which we are not able to guarantee a geometrical decrease of the dual gap.

For away steps in which $\gamma_t < \gamma_{\max}$, the away atom is not removed from the current representation of the iterate. Hence $\mathcal{S}_{t+1} = \mathcal{S}_t$, $\alpha_v^{(t+1)} = (1 + \gamma_t)\alpha_v^{(t)}$ for $v \in \mathcal{S}_t \setminus \{v_t\}$ and $\alpha_{v_t}^{(t+1)} = (1 + \gamma_t)\alpha_{v_t}^{(t)} - \gamma_t$ otherwise.

Note that when choosing Away step in Line 11, it cannot happen that $\alpha_{v_t} = 1$. Indeed this would imply $x_t = v_t$, and so $d_t^A = 0$. Since we would have $\mathcal{S}_t = \{v_t\}$ and the LMO of Line 4 is performed over $\mathcal{S}_t \cup \mathcal{A}_t$, we necessarily have $\langle -\nabla f(x_t), d_t^{\text{FW}} \rangle \geq 0$. It thus leads to a choice of FW step, contradiction.

Algorithm 11 Randomized Away-steps FW (RAFW)

Input: $x_0 \in \mathcal{C}$, $x_0 = \sum_{v \in \mathcal{A}} \alpha_v^{(0)} v$ with $|\mathcal{S}_0| = s$, a subsampling parameter $1 \leq p \leq |\mathcal{A}|$.

1:
2: **for** $t = 0, 1 \ldots, T$ **do**
3: Get $\mathcal{A}_t$ by sampling $\min\{p, |\mathcal{A} \backslash \mathcal{S}_t|\}$ elements uniformly from $\mathcal{A} \backslash \mathcal{S}_t$.
4: Compute $s_t = \text{LMO}(\nabla f(x_t), \mathcal{S}_t \cup \mathcal{A}_t)$
5: Let $d_t^{\text{FW}} = \boldsymbol{s}_t - x_t$ ▷ RFW direction
6: Compute $\boldsymbol{v}_t = \text{LMO}(-\nabla f(x_t), \mathcal{S}_t)$
7: Let $d_t^A = x_t - v_t$. ▷ Away direction
8: **if** $\langle -\nabla f(x_t), d_t^{\text{FW}} \rangle \geq \langle -\nabla f(x_t), d_t^A \rangle$ **then**
9: $d_t = d_t^{\text{FW}}$ and $\gamma_{\max} = 1$ ▷ FW step
10: **else**
11: $d_t = d_t^A$ and $\gamma_{\max} = \alpha_{v_t}^{(t)} / (1 - \alpha_{v_t}^{(t)})$ ▷ Away step
12: **end if**
13: Set γ_t by line-search, with $\gamma_t = \text{argmax}_{\gamma \in [0, \gamma_{\max}]} f(x_t + \gamma d_t)$
14: Let $x_{t+1} = x_t + \gamma_t d_t$ ▷ update $\alpha^{(t+1)}$ (see text)
15: Let $\mathcal{S}_{t+1} = \{v \in \mathcal{A} \text{ s.t. } \alpha_v^{(t+1)} > 0\}$
16: **end for**

Output:

Per iteration cost. Establishing the per iteration cost of this algorithm is not as straightforward as for RFW, as the cost of some operations depends on the size of the active set, which varies throughout the iterations. However, for problems with sparse solutions, we have observed empirically that the size of the active set remains small, making the cost of the second LMO and the comparison of Line 8 negligible compared to the cost of an LMO over the full atomic domain. In this regime, and assuming that the atomic domain has a sparse structure that allows gradient coordinate subsampling, RAFW can achieve a per iteration cost that is, like RFW, roughly $|\mathcal{A}|/p$ times lower than that of its deterministic counterpart.

4.3.1 Analysis

We now provide a convergence analysis of the Randomized Away-steps FW algorithm. These convergence results are stated in terms of the away curvature constant C_f^A and the geometric strong convexity μ_f^A, which are described in 4.B and in [Lacoste-Julien and Jaggi, 2015b]. Throughout this section we assume that f has bounded C_f^A, which is implied by the usual assumption of Lipschitz continuity of the gradient, and strictly positive geometric strong convexity constant μ_f^A, which is verified whenever f is strongly convex and the domain is a polytope.

Theorem 4.3.1. *Let $\mathcal{C} = \text{conv}(\mathcal{A})$, with $\mathcal{A}$ a finite set of extreme atoms. Then after T iterations of Algorithm 11 (RAFW) we have the following linear convergence rate*

$$\mathbb{E}[h(x_{T+1})] \leq (1 - \eta^2 \rho_f)^{\max\{0, \lfloor (T-s)/2 \rfloor\}} h(x_0) \ , \tag{4.3}$$

with $\rho_f = \frac{\mu_f^A}{4 C_f^A}$, $\eta = \frac{p}{|\mathcal{A}|}$ and $s = |\mathcal{S}_0|$.

Proof. See 4.B. ∎

Proof sketch. Our proof structure roughly follows that of the deterministic case in [Lacoste-Julien and Jaggi, 2015b, Beck and Tetruashvili, 2013] with some key differences due to the LMO randomness, and can be decomposed into three parts.

The *first part* consists in upper bounding h_t and is no different from the proof of its deterministic counterpart [Lacoste-Julien and Jaggi, 2015b, Beck and Tetruashvili, 2013].

The *second part* consists in lower bounding the progress $h_t - h_{t+1}$. For this algorithm we can guarantee a decrease of the form

$$h_{t+1} \leq h_t(1 - \rho_f(\frac{g_t}{\tilde{g}_t})^2)^{z_t} \ , \tag{4.4}$$

where $g_t = \langle -\nabla f(x_t), s_t - v_t \rangle$ is the *partial pair-wise dual gap* while $\tilde{g}_t$ is the *pair-wise dual gap*, in which s_t is replaced by the result of a full (and not subsampled) LMO.

We can guarantee a geometric decrease in expectation on h_t at each iteration, except for bad drop steps, where we can only secure $h_{t+1} \leq h_t$. We mark these by setting $z_t = 0$.

One crucial issue is then to quantify $g_t/\tilde{g}_t$. This can be seen as a measure of the quality of the subsampled oracle: if it selects the same atom as the non-subsampled oracle the quotient will be 1, in all other cases it will be ≤ 1.

To ensure a geometrical decrease we further study the probability of events $z_t = 1$ and $\widetilde{g}_t = g_t$: first, we produce a simple bound on the number of bad drop steps (where $z_t = 0$). Second, when $z_t = 1$ holds, Lemma 4.B.3 provides a lower bound on the probability of $g_t = \widetilde{g}_t$.

The *third and last part* of the proof analyzes the expectation of the decrease rate $\prod_{t=0}^{T}(1 - \rho_f(\frac{g_t}{\tilde{g}_t})^2)^{z_t}$ given the above discussion. We produce a conservative bound assuming the maximum possible number of bad drop steps. The key element in this part is to make this maximum a function of the size of the support of the initial iterate and of the number of iteration. The convergence bound is then proven by induction. ■

Comparison with deterministic convergence rates. The rate for away Frank-Wolfe in [Lacoste-Julien and Jaggi, 2015b, Theorem 8], after T iteration is

$$h(x_{T+1}) \leq (1 - \rho_f)^{\lfloor T/2 \rfloor} h(x_0) \ . \tag{4.5}$$

Due to the dependency on η^2 of the convergence rate in Theorem 4.3.1, our bound does not show that RAFW is computationally more efficient than AFW. Indeed we use a very conservative proof technique in which we measure progress only when the sub-sampling oracle equals the full one. Also, the cost of both LMOs depends on the support of the iterates which is unknown a priori except for a coarse upper bound (e.g. the support cannot be more than the number of iterations). Nevertheless, the numerical results do show speed ups compared to the deterministic method.

Beyond strong convexity. The strongly convex objective assumption may not hold for many problem instances. However, the linear rate easily holds for f of the form $g(\boldsymbol{A}x)$ where g is strongly convex and $\boldsymbol{A}$ a linear operator. This type of functions are commonly known as a $\tilde{\mu}$-generally strongly convex function Beck and Tetruashvili [2013], Wang and Lin [2014] or Lacoste-Julien and Jaggi [2015b] (see "Away curvature and geometric strong convexity" in 4.B for definition). The proof simply adapts that of [Lacoste-Julien and Jaggi, 2015b, Th. 11] to our setting.

Theorem 4.3.2. *Suppose f has bounded smoothness constant C_f^A and is $\tilde{\mu}$-generally-strongly convex. Consider the set $\mathcal{C} = \text{conv}(\mathcal{A})$, with $\mathcal{A}$ a finite set of extreme atoms. Then after T iterations of Algorithm 11, with $s = |\mathcal{S}_0|$ and a p parameter of sub-sampling, we have*

$$\mathbb{E}\big[h(x_{T+1})\big] \leq (1 - \eta^2 \tilde{\rho}_f)^{\max\{0, \lfloor \frac{T-s}{2} \rfloor\}} h(x_0) \ , \tag{4.6}$$

with $\tilde{\rho}_f = \frac{\tilde{\mu}}{4C_f^A}$ and $\eta = \frac{p}{|\mathcal{A}|}$.

Proof. See end of 4.B. ■

4.4 Applications

In this section we compare the proposed methods with their deterministic versions. We consider two regularized least squares problems: one with ℓ_1 regularization and another one with latent group lasso (LGL) regularization. In the first case, the domain is a polytope and the analysis of AFW and RAFW holds.

Our results show the FW gap versus number of iterations, and also cumulative number of computed gradient coefficients, which we will label *"nbr coefficients of grad"*. This allows to better reflect the true complexity of our experiments since sub-sampling the LMO in the problems we consider amounts to computing the gradient on a subset of coordinates.

In the case of latent group lasso, we also compared the performance of RFW against FW in terms of wall-clock time on a large dataset stored in disk and accessed sequentially in chunks (i.e., in streaming model).

4.4.1 Lasso problem

Synthetic dataset. We generate a synthetic dataset following the setting of Lacoste-Julien and Jaggi [2015b], with a Gaussian design matrix A of size $(200, 500)$ and noisy measurements $b = Ax^* + \boldsymbol{\varepsilon}$, with $\boldsymbol{\varepsilon}$ a random Gaussian vector and x^* a vector with 10% of nonzero coefficients and values in $\{-1, +1\}$.

In Figures 4.1 and 4.2, we consider a problem of the form (OPT), where the domain is an ℓ_1 ball, a problem often referred to as Lasso. We compare FW against RFW, and AFW against RAFW. The ℓ_1 ball radius set to 40, so that the unconstrained optimum lies outside the domain.

RFW experiments. Figure 4.1 shows a comparison between FW and RFW. Each call to the randomized LMO outputs a direction, likely less aligned with the opposite of the gradient than the direction proposed by FW, which explains why RFW requires more iterations to converge on the upper left graph of Figure 4.1. Each call of the randomized LMO is cheaper than the LMO in terms of number of computed coefficients of the gradient, and the trade-off is beneficial as can be seen on the bottom left graph, where RFW outperforms its deterministic variant in terms of *nbr coefficients of grad*.

Finally, the right panels of Figure 4.1 provide an insight on the evolution of the sparsity of the iterate, depending on the algorithm. FW and RFW perform similarly in terms of the fraction of recovered support (bottom right graph). In terms of the sparsity of the iterate, RFW under-performs FW (upper right graph). This can be explained as follows: because of the sub-sampling, each atom of the randomized LMO provides a direction less aligned with the

opposite of the gradient than the one provided by the LMO. Each update in such a direction may result in putting weight on an atom that would better be off the representation of the iterate. It impacts the iterate all along the algorithm as vanilla FW removes past atoms from the representation only by multiplicatively shrinking their weight.

RAFW experiments. Unlike RFW, the RAFW method outperforms AFW in terms on number of iterations in the upper left graph in Figure 4.2. These graphs also illustrate the linear rate of convergence of both algorithms. The bottom left graph shows that the gap between RAFW and AFW is even larger when comparing the cumulative number of computed coefficients of the gradient required to reach a certain target precision.

This out-performance of RAFW over AFW in term of number of iteration to converge is not predicted by our convergence analysis. We conjecture that the away mechanism improves the trade-off between the cost of the LMO and the alignment of the descent direction with the opposite of the gradient. Indeed, because of the oracle subsampling, the partial FW gap (e.g. the scalar product of the Randomized FW direction with the opposite of the gradient) in RAFW is smaller than in the non randomized variant, and so there is a higher likelihood of performing an away step.

Finally, the away mechanism enables the support of the RAFW to stay close to that of AFW, which was not the case in the comparison of RFW versus FW. This is illustrated in the right panels of Figure 4.2.

Real dataset. On figure 4.3, we test the Lasso problem on the E2006-tf-idf data set [Kogan et al., 2009], which gathers volatility of stock returns from companies with financial reports. Each financial reports is then represented through its TF-IDF embedding ($n = 16087$ and $d = 8000$ after an initial round of feature selection). The regularizing parameter is chosen to obtain solution with a fraction of 0.01 nonzero coefficients.

4.4.2 Latent Group-Lasso

Notation. We denote by $[d]$ the set of indices from 1 to d. For $g \subseteq [d]$ and $x \in \mathbb{R}^d$, we denote by $x_{(g)}$ the projection of x onto the coordinates in g. We use the notation $\nabla_{(g)} f(x_t)$ to denote the gradient with respect to the variables in group g. Similarly $x_{[g]} \in \mathbb{R}^d$ is the vector that equals x in the coordinates of g and 0 elsewhere.

Model. As outlined by Jaggi [2013], FW algorithms are particularly useful when the domain is a ball of the latent group norm [Obozinski et al., 2011]. Consider a collection $\mathcal{G}$ of subsets of $[d]$ such that $\bigcup_{g\in\mathcal{G}} g = [d]$ and denote by $||\cdot||_g$ any norm on $\mathbb{R}^{|g|}$. Frank-Wolfe can be used to solve (OPT) with $\mathcal{C}$ being the ball corresponding to the latent group norm

$$||x||_{\mathcal{G}} \stackrel{\text{def}}{=} \min_{v_{(g)}\in\mathbb{R}^{|g|}} \sum_{g\in\mathcal{G}} ||v_{(g)}||_g \quad s.t. \;\; x = \textstyle\sum_{v\in\mathcal{G}} v_{[g]} \; . \tag{4.7}$$

This formulation matches a constrained version of the regularized [Obozinski et al., 2011, equation (5)] when each $||\cdot||_g$ is proportional to the Euclidean norm. For simplicity we will consider $||\cdot||_g$ to be the euclidean norm.

When $\mathcal{G}$ forms a partition of $[d]$ (i.e., there is no overlap between groups), this norm coincides with the group lasso norm.

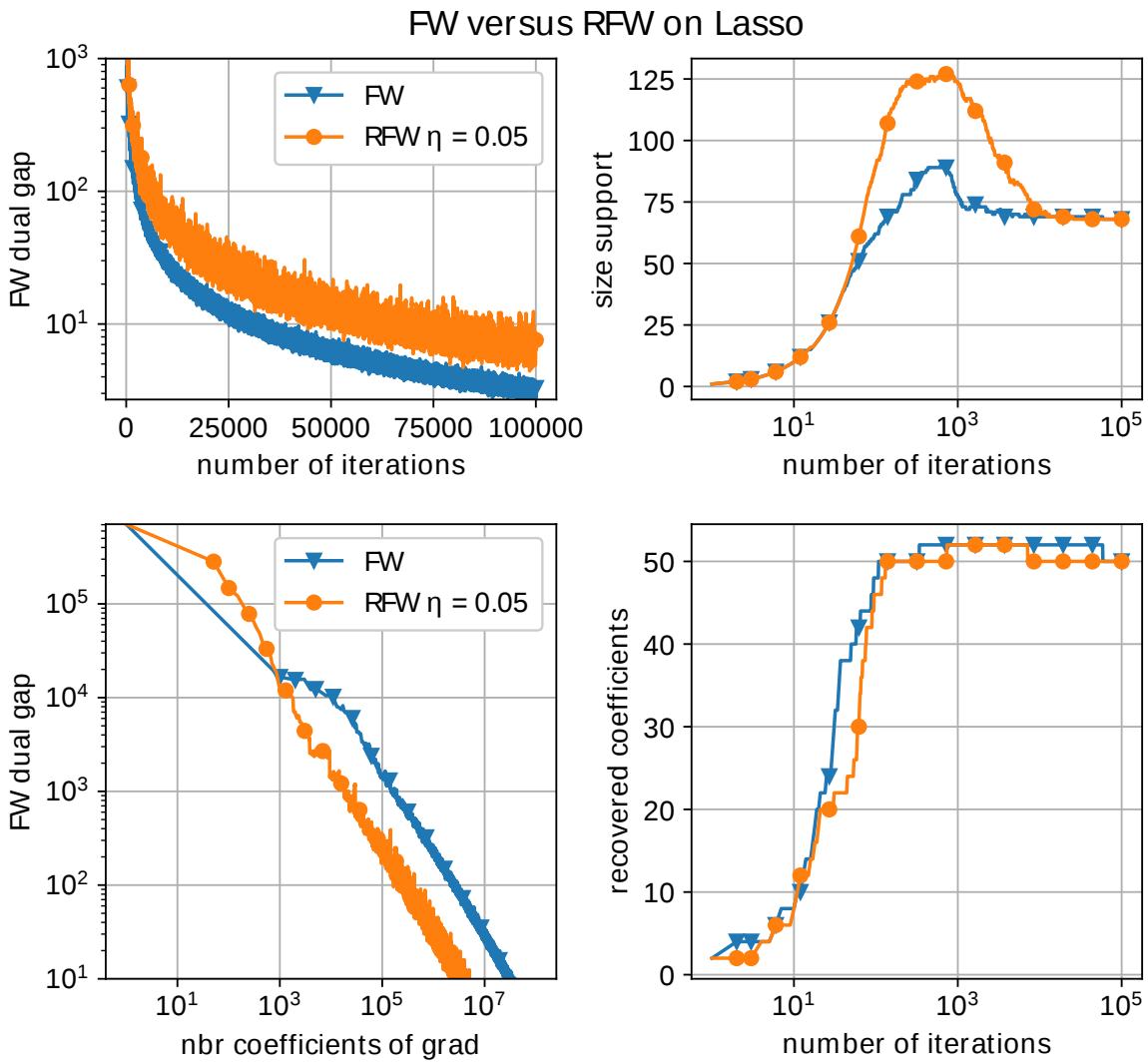


Figure 4.1: Comparison between FW and RFW with subsampling parameter $\eta = \frac{p}{|\mathcal{A}|} = 0.05$ (chosen arbitrarily) on the lasso problem. *Upper left:* progress in FW dual gap versus number of iterations. *Lower left:* progress of the FW dual gap versus cumulative number of computed coefficients of gradient per call to LMO, called *nbr coefficients of grad* here. *Lower right:* recovered coefficients in support of the ground truth versus number of iterations. *Upper right*: size of support of iterate versus number of iterations.

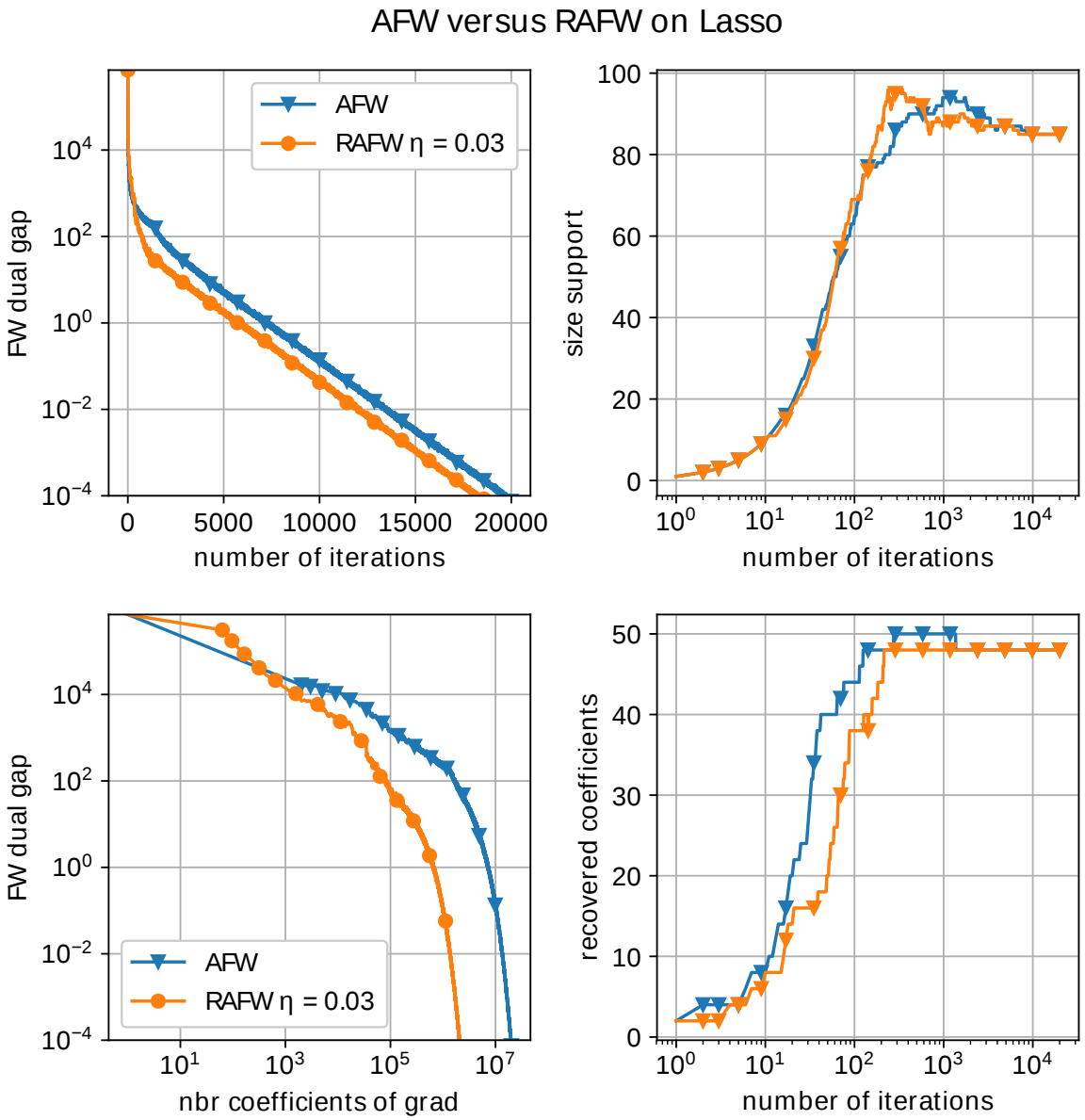


Figure 4.2: Same parameters and setting as in Figure 4.1 but to compare RAFW and AFW. AFW performed 880 away steps among which 14 were a drop steps while RAFW performed 1242 away steps and 37 drop steps.

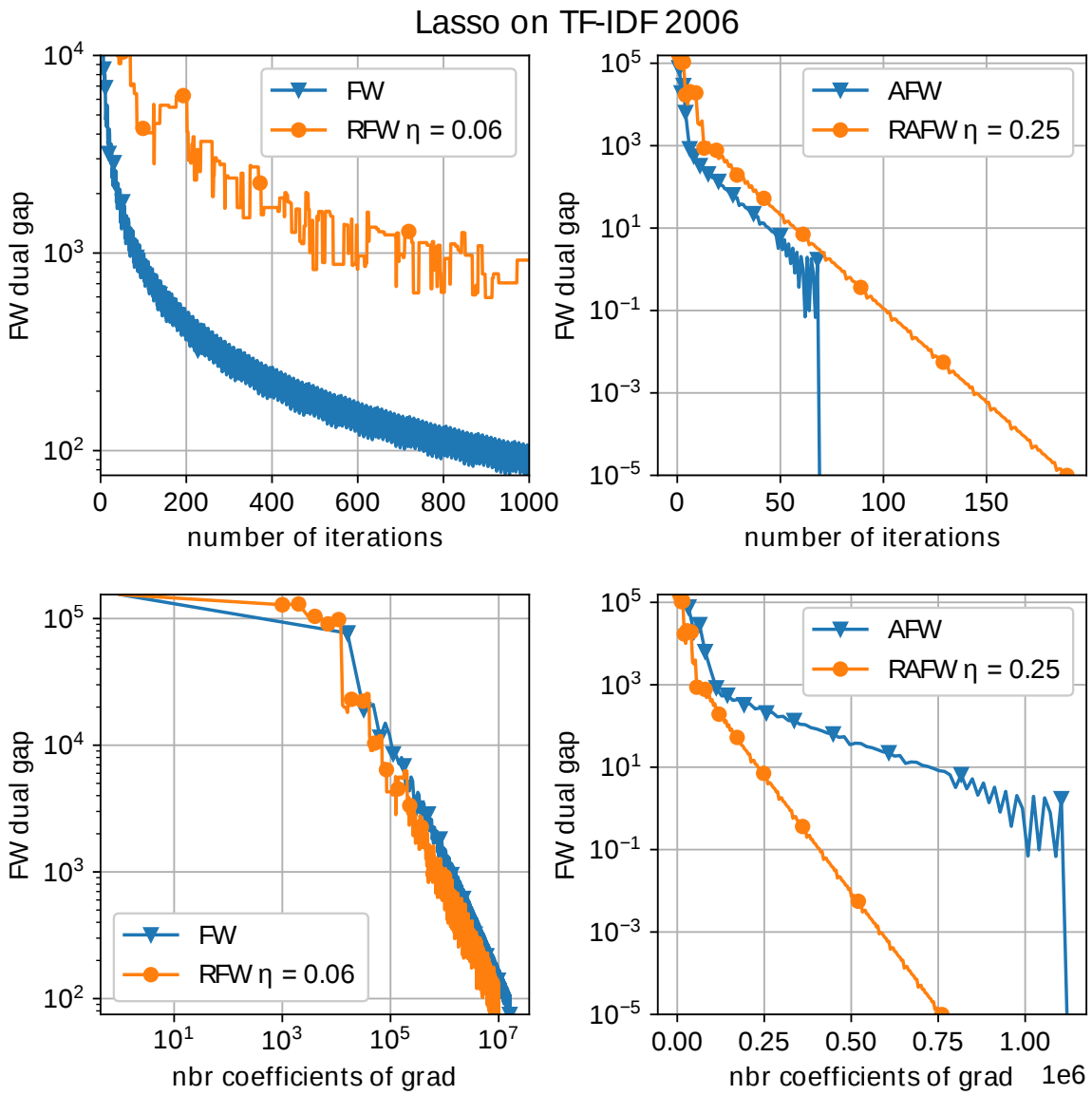


Figure 4.3: Performance of FW and AFW against RFW and RAFW respectively on the lasso problem with TF-IDF 2006 dataset. The subsampling parameter is $\eta = \frac{p}{|\mathcal{A}|} = 0.06$ (again chosen arbitrarily) for RFW and $\eta = 0.25$ for RAFW. *Right:* Comparison of RAFW against RFW. *Left:* Comparison of RFW against FW. *Upper:* progress in FW dual gap versus number of iterations. *Lower:* progress of the FW dual gap versus cumulative number of computed coefficients in gradient per call to LMO.

Sub-sampling. Given $g \in \mathcal{G}$, consider the hyper-disk

$$\mathbb{D}_g(\beta) = \left\{ v \in \mathbb{R}^d \;\middle|\; v = v_{[g]}, ||v_{(g)}|| \leq \beta \right\} .$$

Obozinski et al. [2011, Lemma 8] shows that such constrain set $\mathcal{C}$ is the convex hull of $\mathcal{A} \stackrel{\text{def}}{=} \bigcup_{g \in \mathcal{G}} \mathbb{D}_g$.

The RFW can be used to solve this problem, with $\mathcal{A}_t \stackrel{\text{def}}{=} \bigcup_{g \in \mathcal{G}_p} \mathbb{D}_g$ and where the random oracle is performed over a random subset $\mathcal{G}_p \subseteq \mathcal{G}$ of size p. Denoting by $g_p = \bigcup_{g \in \mathcal{G}_p} g$ the LMO in RFW becomes

$$\text{LMO}(x_t, \mathcal{A}_t) \in \operatorname*{argmax}_{v \in \mathcal{A}_t} \langle v_{(g_p)}, -\nabla_{(g_p)} f(x_t) \rangle .$$

With this formulation we only need to compute the gradient on the g_p index. Depending on $\mathcal{G}$ and on the sub-sampling rate, this can be a significant computational benefit.

Experiments. We illustrate the convergence speed-up of using RFW over FW for latent group lasso regularized least square regression.

For $d = 10000$ we consider a collection $\mathcal{G}$ of groups of size 10 with an overlap of 3 and the associated atomic set $\mathcal{A}$. We chose the ground truth parameter vector $w_0 \in \text{conv}(\mathcal{A})$ with a fraction of 0.01 of nonzero coefficients, where on each active group, the coefficients are generated from a Gaussian distribution. The data is a set of n pairs $(y_i, w_i) \in \mathbb{R} \times \mathbb{R}^d$, where w_i is generated from a Gaussian distribution and $y_i = w_i^T w_0 + \varepsilon_i$, where ε_i is again a Guassian random variable. The regularizing parameter is $\beta = 14$, set so that the unconstrained optimum lies outside of the constrain set.

Large dataset and Streaming Model. The design matrix is stored in disk. We allow both RFW and FW to access it only through chunks of size $n \times 500$. This streaming model allows a wall clock comparison of the two methods on very large scale problems.

Computing the gradient when the objective is the least squares loss consists in a matrix vector product. Computing it on a batch of coordinates then requires same operation with a smaller matrix. When computing the gradient at each randomized LMO call, the cost of slicing the design matrix can then compensate the gain in doing a smaller matrix vector product.

With data loaded in memory, which is typically the case for large datasets, both the LMO and the randomized LMO have this access data cost. Consider also that RFW allows any scheme of sampling, including one that minimizes the cost of data retrieval.

4.5 Conclusion

We gave theoretical guarantees of convergence of randomized versions of FW that exhibit same order of convergence as their deterministic counter-parts. As far as we know, for the case of RAFW, this is the first contribution of the kind. While the theoretical complexity bounds don't necessarily imply this, our numerical experiments show that randomized versions often outperform their deterministic ones on ℓ_1-regularized and latent group lasso regularized least squares. In both cases, randomizing the LMO allows us to compute the gradient only on a

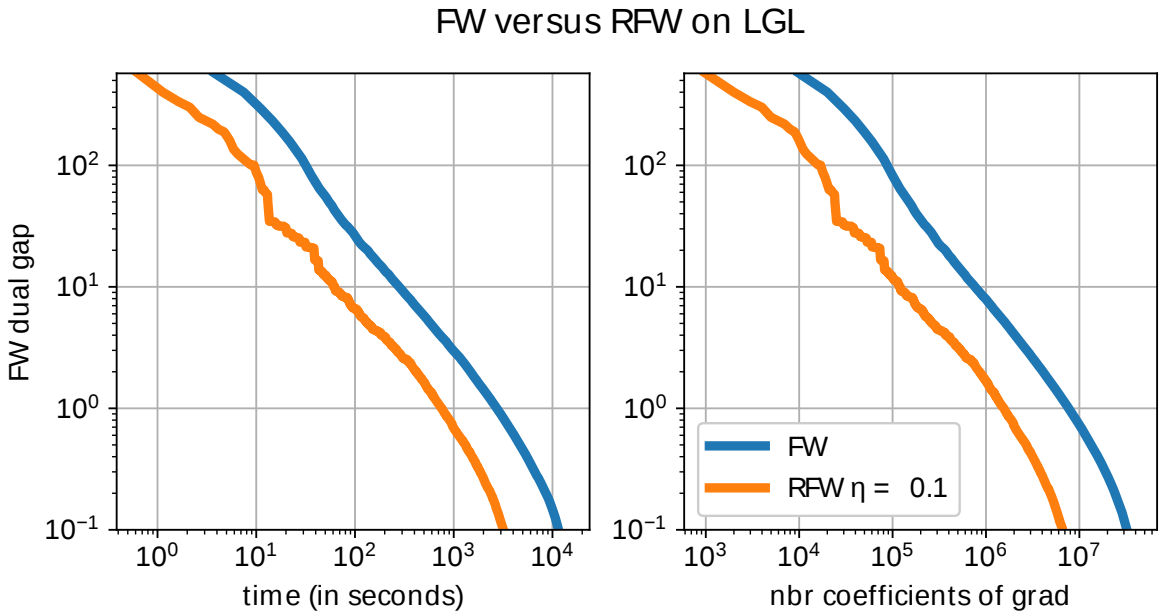


Figure 4.4: Both panels are in log log scale and show convergence speed up for FW and RFW on latent group lasso regularized least square regression. The parameter of subsampling $\eta = 0.1$, is chosen arbitrarily. *Left:* evolution of the precision in FW dual gap versus the wall clock time. *Right:* evolution of the precision in FW dual gap versus the cumulative number of computed coefficients of the gradient.

subset of its coordinates. We used it to speed up the method in a streaming model where the data is accessed by chunks, but there might be other situations where the structure of the polytope can be leveraged to make subsampling computationally beneficial.

There are other linearly-convergent variants of FW besides AFW, such as the Pairwise FW algorithm [Lacoste-Julien and Jaggi, 2015b], for which it might be possible to derive randomized variants.

Finally, recent results such as [Goldfarb et al., 2016, 2017, Hazan and Luo, 2016] combine various improvements of FW (away mechanism, sliding, lazy oracles, stochastic FW, etc.). Randomized oracles add to this toolbox and could further improve its benefits.

Appendices

Appendix notations. We denote by $\mathbf{E}_t$ the conditional expectation at iteration t, conditioned on all the past and by $\mathbb{E}$ a full expectation. We denote by a tilde the values that come from the deterministic analysis of FW. Denote by $r_t = -\nabla f(x_t)$. For $k \in \mathbb{N}^*$, denote by $[k]$ all integer between 1 and k.

4.A Proof of Subsampling for Frank-Wolfe

In this section we provide a convergence proof for Algorithm 10. The proof is loosely inspired by that of [Locatello et al., 2017a, Appendix B.1], with the obvious difference that the result of the LMO is a random variable in our case.

Theorem 4.2.1′. *Let f be a function with bounded curvature constant C_f, Algorithm 10 for $\eta \in (0, 1]$, (with step-size chosen by either variants) converges towards a solution of (OPT), satisfying*

$$\mathbb{E}(f(x_T)) - f(x^*) \leq \frac{2(C_f + f(x_0) - f(x^*))}{\eta T + 2} \ . \tag{4.8}$$

Proof. By definition of the curvature constant, at iteration t we have

$$f(x_t + \gamma(s_t - x_t)) \leq f(x_t) + \gamma\langle \nabla f(x_t), s_t - x_t\rangle + \frac{\gamma^2}{2} C_f \ . \tag{4.9}$$

By minimizing with respect to γ on $[0, 1]$ we obtain

$$\gamma_t = \text{clip}_{[0,1]}\langle -\nabla f(x_t), s_t - x_t\rangle / C_f \quad , \tag{4.10}$$

which is the definition of γ_t in the algorithm with Variant 2. Hence, we have

$$f(x_{t+1}) \quad \leq \quad f(x_t) + \min_{\gamma \in [0,1]} \left\{ \gamma\langle \nabla f(x_t), s_t - x_t\rangle + \frac{\gamma^2}{2} C_f \right\},$$

an inequality which is also valid for Variant 1 since by the line search procedure the objective function at x_{t+1} is always equal or smaller than that of Variant 1. Denote by $h_t = f(x_t) - f(x^*)$,

$$h_{t+1} \leq h_t + \min_{\gamma \in [0,1]} \left\{ \gamma\langle \nabla f(x_t), s_t - x_t\rangle + \frac{\gamma^2}{2} C_f \right\}.$$

We write $\widetilde{s}_t$ the FW atom if we had started the FW algorithm at x_t, and $\mathbf{E}_t$ the expectation conditionned on all the past until x_t, we have

$$\mathbf{E}_t h_{t+1} \leq h_t + \mathbf{E}_t \min_{\gamma \in [0,1]} \left\{ \gamma\langle \nabla f(x_t), s_t - x_t\rangle + \frac{\gamma^2}{2} C_f \right\} \tag{4.11}$$

$$\leq h_t + \mathcal{P}(s_t = \widetilde{s}_t) \min_{\gamma \in [0,1]} \left\{ \gamma\langle \nabla f(x_t), \widetilde{s}_t - x_t\rangle + \frac{\gamma^2}{2} C_f \right\} \tag{4.12}$$

$$\leq h_t + \eta \min_{\gamma \in [0,1]} \left\{ -\gamma h(x_t) + \frac{\gamma^2}{2} C_f \right\} \tag{4.13}$$

$$\leq h_t + \eta(-\gamma h(x_t) + \frac{\gamma^2}{2} C_f) \quad \text{(for any } \gamma \in [0, 1], \text{ by definition of min)} \ , \tag{4.14}$$

where the second inequality follows from the definition of expectation and the fact that minimum is non-positive since it is zero for $\gamma = 0$. The last inequality is a consequence of uniform sampling as well as it uses that the FW gap is an upper bound on the dual gap, e.g. $\langle -\nabla f(x_t), \widetilde{s}_t - x_t \rangle \geq h(x_t)$.

Induction. From (4.14) the following is true for any $\gamma \in [0, 1]$

$$\mathbf{E}_t(h_{t+1}) \leq h_t(1 - \eta\gamma) + \frac{\gamma^2}{2}\eta C_f \ . \tag{4.15}$$

Taking unconditional expectation and writing $H_t = \mathbb{E}(h_t)$, we get for any $\gamma \in [0, 1]$

$$H_{t+1} \leq H_t(1 - \eta\gamma) + \frac{\gamma^2}{2}\eta C_f. \tag{4.16}$$

With $\gamma_t = \frac{2}{\eta t+2} \in [0, 1]$, we get by induction

$$H_t \leq 2\frac{C_f + \epsilon_0}{\eta t + 2} = \gamma_t(C_f + \epsilon_0), \tag{4.17}$$

where $\epsilon_0 = f(x_0) - f(x^*)$. Initialization follows the fact that the curvature constant is positive. For $t > 0$, from (4.16) and the induction hypothesis

$$\begin{aligned}
H_{t+1} &\leq \gamma_t(C_f + \epsilon_0)(1 - \eta\gamma_t) + \frac{\gamma_t^2}{2}\eta C_f \\
&\leq \gamma_t(C_f + \epsilon_0)(1 - \eta\gamma_t) + \frac{\gamma_t^2}{2}\eta(C_f + \epsilon_0) \\
&\leq \gamma_t(C_f + \epsilon_0)(1 - \eta\gamma_t + \frac{\gamma_t}{2}\eta) \\
&\leq (C_f + \epsilon_0)(1 - \frac{\gamma_t}{2}\eta)\gamma_t \\
&\leq (C_f + \epsilon_0)\gamma_{t+1}.
\end{aligned}$$

The last inequality comes from the fact that $(1 - \frac{\gamma_t}{2}\eta)\gamma_t \leq \gamma_{t+1}$. Indeed, with $\gamma_t = \frac{2}{\eta t+2}$, it is equivalent to

$$\begin{aligned}
(1 - \frac{\eta}{\eta t + 2})\frac{2}{\eta t + 2} &\leq \frac{2}{\eta(t+1) + 2} \\
\Leftrightarrow \frac{(\eta t + 2) - \eta}{\eta t + 2} &\leq \frac{\eta t + 2}{\eta(t+1) + 2} \\
\Leftrightarrow (\eta t + 2 - \eta)(\eta(t+1) + 2) &\leq (\eta t + 2)^2 \\
\Leftrightarrow \eta^2 t^2 + 4\eta t + 4 - \eta^2 &\leq \eta^2 t^2 + 4\eta t + 4.
\end{aligned}$$

The last being true, it concludes the proof. ■

4.B Proof of Subsampling for Away-steps Frank-Wolfe

Away curvature and geometric strong convexity. The *away curvature* constant is a modification of the curvature constant described in the previous subsection, in which the FW direction $s - x$ is replaced with an arbitrary direction $s - v$:

$$C_f^A \triangleq \sup_{\substack{x,s,v\in\mathcal{C}\\ \gamma\in[0,1]\\ y=x+\gamma(s-v)}} \frac{2}{\gamma^2}\big(f(y) - f(x) - \gamma\langle\nabla f(x), s - v\rangle\big) \ .$$

The *geometric strong convexity* constant μ_f depends on both the function and the domain (in contrast to the standard strong convexity definition) and is defined as (see "An Affine Invariant Notion of Strong Convexity" in [Lacoste-Julien and Jaggi, 2015b] for more details)

$$\mu_f^A = \inf_{x\in\mathcal{C}} \inf_{\substack{x^*\in\mathcal{C}\\ \langle\nabla f(x), x^*-x\rangle<0}} \frac{2}{\gamma^A(x,x^*)^2} B_f(x, x^*)$$

where $B_f(x, x^*) = f(x^*) - f(x) - \langle\nabla f(x), x^* - x\rangle$ and $\gamma^A(x, x^*)$ the positive step-size quantity:

$$\gamma^A(x, x^*) \coloneqq \frac{\langle -\nabla f(x), x^* - x\rangle}{\langle -\nabla f(x), s_f(x) - v_f(x)\rangle}.$$

In particular $s_f(x)$ is the Frank Wolfe atom starting from x. $v_f(x)$ is the away atom when considering all possible expansions of x as a convex combinations of atoms in $\mathcal{A}$. Denote by $\mathcal{S}_x \coloneqq \{\mathcal{S} \mid \mathcal{S} \subseteq \mathcal{A}$ such that x is a proper convex combination of all elements in $\mathcal{S}\}$ and by $v_{\mathcal{S}(x)} \coloneqq \operatorname{argmax}_{v\in\mathcal{S}}\langle\nabla f(x), v\rangle$. $v_f(x)$ is finally defined by

$$v_f(x) \triangleq \operatorname*{argmin}_{\{v=v_{\mathcal{S}}|\mathcal{S}\in\mathcal{S}_x\}} \langle\nabla f(x), v\rangle \ .$$

Following [Lacoste-Julien and Jaggi, 2015b, Lemma 9 in Appendix F], the geometric $\tilde{\mu}$-generally-strongly-convex constant is defined as

$$\tilde{\mu}_f = \inf_{x\in\mathcal{C}} \inf_{\substack{x^*\in\chi^*\\ \langle\nabla f(x), x^*-x\rangle<0}} \frac{1}{2\gamma^A(x,x^*)^2}\big(f(x^*) - f(x) - 2\langle\nabla f(x), x^* - x\rangle\big) \ ,$$

where χ^* represents the solution set of (OPT).

Notations. In the context of RAFW, $\mathcal{A}$ denotes the finite set of extremes atoms such that $\mathcal{C} = \operatorname{Conv}(\mathcal{A})$. At iteration t, $\mathcal{A}_t$ is a random subset of element of $\mathcal{A} \setminus \mathcal{S}_t$ where $\mathcal{S}_t$ is the current support of the iterate. The Randomized LMO is performed over $\mathcal{V}_t = \mathcal{S}_t \cup \mathcal{A}_t$ so that for Algorithm 11, $s_t \overset{\text{def}}{\in} \operatorname{argmax}_{v\in\mathcal{V}_t}\langle -\nabla f(x_t), v\rangle$ is the FW atom at iteration t for RAFW.

Note that when $|\mathcal{A} \setminus \mathcal{S}_t| \leq p$, Algorithm 11 does exactly the same as AFW. For the sake of simplicity we will consider that this is not the case. Indeed we would otherwise fall back into the deterministic setting and the proof would just be that of Lacoste-Julien and Jaggi [2015b].

We use tilde notation for quantities that are specific to the deterministic FW setting. For instance, $\widetilde{s}_t \overset{\text{def}}{\in} \operatorname{argmax}_{v\in\mathcal{A}}\langle -\nabla f(x_t), v\rangle$ is the FW atom for AFW starting at x_t.

Similarly the Away atom is such that $v_t \overset{\text{def}}{\in} \operatorname{argmin}_{v\in\mathcal{S}_t}\langle -\nabla f(x_t), v\rangle$ and it does not depend on the sub-sampling at iteration t. Here we do not use any tilde because it is a quantity that appears both in AFW and its Randomized counter-part.

In AFW, $\widetilde{g}_t \triangleq \langle -\nabla f(x_t), \widetilde{s}_t - v_t \rangle = \max_{s \in \mathcal{A}} \langle -\nabla f(x_t), s - v_t \rangle$ is an upper-bound of the dual gap, named the *pair-wise dual gap* [Lacoste-Julien and Jaggi, 2015b]. We consider the corresponding *partial pair-wise dual gap* $\widetilde{g}_t \triangleq \langle -\nabla f(x_t), s_t - v_t \rangle = \max_{s \in \mathcal{V}_t} \langle -\nabla f(x_t), s - v_t \rangle$. It is partial is the sense that the maximum is computed on a subset $\mathcal{V}_t$ of $\mathcal{A}$ which results in the fact that it is not guaranteed anymore to be an upper-bound on the dual-gap.

Structure of the proof. The proof is structured around a main part that uses Lemmas 4.B.1 and 4.B.3. Lemma 4.B.2 is only used to prove Lemma 4.B.3.

The main proof follows the scheme of the deterministic one of AFW in [Lacoste-Julien and Jaggi, 2015b, Theorem 8]. It is divided in three parts. The first part consists in upper bounding $h_t \stackrel{\text{def}}{=} f(x_t) - f(x^*)$ with $\tilde{g}_t$. It does not depend on the specific construction of the iterates x_t and thus remains the same as that in Lacoste-Julien and Jaggi [2015b]. The second part provides a lower bound on the progress on the algorithm, namely

$$h_{t+1} \leq (1 - \rho_f (\frac{g_t}{\widetilde{g}_t})^2) h_t, \tag{4.18}$$

with $\rho_f = \frac{\mu_f^A}{4C_f^A}$, when it is not doing a *bad drop step* (defined above). As a proxy for this event, we use the binary variable z_t that equals 0 for bad drop steps and 1 otherwise.

The difficulty lies in the fact that we guarantee a geometrical decrease only when $g_t = \widetilde{g}_t$ and $z_t = 1$. Because of the sub-sampling and unlike in the deterministic setting, z_t is a random variable. Lemma 4.B.3 provides a lower bound on the probability of interest, $\mathcal{P}(\widetilde{g}_t = g_t \mid z_t = 1)$, for the last part of the main proof.

Finally, the last part of the proof constructs a bound on the number of times we can expect both $z_t = 1$ and $g_t = \widetilde{g}_t$ subject to the constraint that at least half of the iterates satisfy $z_t = 1$. It is done by recurrence.

4.B.1 Lemmas

This lemma ensures the chosen direction d_t in RAFW is a good descent direction, and links it with g_t which may be equal to $\widetilde{g}_t$.

Lemma 4.B.1. *Let s_t, v_t and d_t be as defined in Algorithm 11. Then for $g_t \stackrel{def}{=} \langle -\nabla f(x_t), s_t - v_t \rangle$, we have*

$$\langle -\nabla f(x_t), d_t \rangle \geq \frac{1}{2} g_t \geq 0 \ . \tag{4.19}$$

Proof. The first inequality appeared already in the convergence proof of Lacoste-Julien and Jaggi [2015b, Eq. (6)], which we repeat here for completeness. By the definition of d_t we have:

$$\begin{aligned} 2\langle -\nabla f(x_t), d_t \rangle &\geq \langle -\nabla f(x_t), d_t^A \rangle + \langle -\nabla f(x_t), d_t^{\text{FW}} \rangle \\ &= \langle -\nabla f(x_t), s_t - v_t \rangle = g_t \end{aligned} \tag{4.20}$$

We only need to prove that g_t is non-negative. In line 4 of algorithm 11, s_t is the output of LMO performs of the set of atoms $\mathcal{S}_t \cup \mathcal{A}_t \triangleq \mathcal{V}_t$,

$$s_t = \operatorname*{argmax}_{s \in \mathcal{V}_t} \langle -\nabla f(x_t), s \rangle \ ,$$

so that we have $\langle -\nabla f(x_t), s_t \rangle \geq \langle -\nabla f(x_t), v_t \rangle$. By definition of g_t, it implies $g_t \geq 0$. ∎

Lemma 4.B.2 is just a simple combinatorial result needed in Lemma 4.B.3. Consider a sequence of m numbers, we lower bound the probability for the maximum of a subset of size greater than p to be equal to the maximum of the sequence.

Lemma 4.B.2. *Consider any sequence $(r_i)_{i\in\mathcal{I}}$ in $\mathbb{R}$ with $\mathcal{I} = \{1, \cdots, m\}$, and a subset $\mathcal{I}_p \subseteq \mathcal{I}$ of size p. We have*

$$\mathcal{P}(\max_{i\in\mathcal{I}_p} r_i = \max_{i\in\mathcal{I}} r_i) \geq \frac{p}{m} \ . \tag{4.21}$$

Proof. Consider $M = \{i \in \mathcal{I} \mid r_i = \max_{j\in\mathcal{I}} r_j\}$. We have $\max_{i\in\mathcal{I}_p} r_i = \max_{i\in\mathcal{I}} r_i$ if and only if at least one element of $\mathcal{I}_p$ belongs to M:

$$\mathcal{P}(\max_{i\in\mathcal{I}_p} r_i = \max_{i\in\mathcal{I}} r_i) = \mathcal{P}(|\mathcal{I}_p \cap M| \geq 1) \ . \tag{4.22}$$

By definition M has at least one element i_0. Since $\{i_0 \in \mathcal{I}_p\} \subset \{|\mathcal{I}_p \cap M| \geq 1\}$

$$\mathcal{P}(|\mathcal{I}_p \cap M| \geq 1) \geq \mathcal{P}(\{i_0 \in \mathcal{I}_p\}) \ . \tag{4.23}$$

All subsets are taken uniformly at random, we just have to count the number of subset $\mathcal{I}_p$ of $\mathcal{I}$ of size p with $i_0 \in \mathcal{I}_p$

$$\mathcal{P}(\{i_0 \in \mathcal{I}_p\}) = \frac{\binom{m-1}{p-1}}{\binom{m}{p}} = \frac{p}{m} \tag{4.24}$$

$$\mathcal{P}(\max_{i\in\mathcal{I}_p} r_i = \max_{i\in\mathcal{I}} r_i) \geq \frac{p}{m} \ . \tag{4.25}$$

■

In the second part of the main proof we ensure a geometric decrease when both $g_t = \widetilde{g}_t$ and $z_t = 1$, i.e. outside of *bad drop steps*. The following lemma helps quantifying the probability of $g_t = \widetilde{g}_t$ holding when $z_t = 1$.

Lemma 4.B.3. *Consider g_t (defined in Lemma 4.B.1) to be the partial pair-wise (PW) dual gap of RAFW at iteration t with sub-sampling parameter p on the constrained polytope $\mathcal{C} = \text{conv}(\mathcal{A})$, where $\mathcal{A}$ is a finite set of extremes points of $\mathcal{C}$. $\widetilde{g}_t \triangleq \max_{s\in\mathcal{A}} \langle -\nabla f(x_t), s - v_t\rangle$ is the pairwise dual gap of AFW starting at x_t on this same polytope. Denote by z_t the binary random variable that equals 0 when the t^{th} iteration of RAFW makes an away step that is a drop step with $\gamma_{max} < 1$ (a bad drop step), and 1 otherwise. Then we have the following bound*

$$\mathcal{P}(g_t = \widetilde{g}_t \mid x_t, z_t = 1) \geq \Big(\frac{p}{|\mathcal{A}|}\Big)^2. \tag{PROB}$$

Proof. Recall that $g_t^A \triangleq \langle r_t, d_t^A\rangle$. By definition $\{z_t = 0\} = \{g_t < g_t^A, \gamma_{\max} < 1, \gamma_t^* = \gamma_{\max}\}$, where $\gamma_t^* \triangleq \text{argmin}_{\gamma\in[0,\gamma_{\max}]} f(x_t + \gamma d_t^A)$. Its complementary $\{z_t = 1\}$ can thus be expressed as the partition $A_1 \cup A_2 \cup A_3$ where the A_i are defined by

$$A_1 = \{g_t \geq g_t^A\} \quad \text{(performs a FW step)} \tag{4.26}$$

$$A_2 = \{g_t < g_t^A \ , \ \alpha_{v_t}^{(t)}/(1 - \alpha_{v_t}^{(t)}) \geq 1\} \quad \text{(performs away step with } \gamma_{\max} \geq 1) \tag{4.27}$$

$$A_3 = \{g_t < g_t^A \ , \ \alpha_{v_t}^{(t)}/(1 - \alpha_{v_t}^{(t)}) < 1 \ , \ \gamma_t^* < \alpha_{v_t}^{(t)}/(1 - \alpha_{v_t}^{(t)})\}. \tag{4.28}$$

First note that in the case of A_2 and A_3, $\gamma_{\max} = \alpha_{v_t}^{(t)}/(1-\alpha_{v_t}^{(t)})$. Though the right hand side formulation highlights that it is entirely determined by x_t, recalling that $\alpha_{v_t}^{(t)}$ is the mass along the atom v_t in the decomposition of x_t in §4.3.

From a higher level perspective, these cases are those for which we can guarantee a geometrical decrease of $h_t = f(x_t) - f(x^*)$ (see second part of main proof). By definition, the A_i are disjoints. A_1 represents a choice of a FW step in RAFW contrary to A_2 and A_3 which stands for an away step choice in RAFW. A_2 is an away step for which there is enough potential mass ($\gamma_{\max} > 1$) to move along the away direction and to ensure sufficient objective decreasing. A_3 encompasses the situations where there is not a lot of mass along the away direction ($\gamma_{\max} < 1$) but which is not a drop step, e.g. the amount of mass is not a limit to the descent.

Our goal is to lower bound $P = \mathcal{P}(g_t = \widetilde{g}_t \mid x_t, z_t = 1)$. The following probabilities will be with respect to the t^{th} sub-sampling only. Notice that g_t^A, $\widetilde{g}_t$ and α_{v_t} are known given $\{x_t, z_t = 1\}$. Using Bayes' rule, and because the A_i are disjoints, we have

$$\begin{aligned} P &= \mathcal{P}(g_t = \widetilde{g}_t \mid x_t, \{z_t = 1\}) \\ &= \frac{\sum_{i=1}^3 \mathcal{P}(g_t = \widetilde{g}_t \mid x_t, A_i)\mathcal{P}(A_i \mid x_t)}{\sum_{i=1}^3 \mathcal{P}(A_i \mid x_t)} \ . \end{aligned} \tag{4.29}$$

By definition of g_t and $\widetilde{g}_t$, $g_t \leq \widetilde{g}_t$, so that measuring the probability of an event like $\{g_t = \widetilde{g}_t\}$ conditionally on $\{g_t \leq g_t^A\}$ will naturally depend on whether or not, the deterministic condition $\widetilde{g}_t \geq g_t^A$ is satisfied. Hence the following case distinction.

Recall $\mathcal{V}_t = \mathcal{S}_t \cup \mathcal{A}_t$.

Case $\widetilde{g}_t < g_t^A$.

$$P = \frac{\sum_{i=1}^3 \mathcal{P}(g_t = \widetilde{g}_t \mid x_t, A_i, \widetilde{g}_t < g_t^A)\mathcal{P}(A_i \mid x_t, \widetilde{g}_t < g_t^A)}{\sum_{i=1}^3 \mathcal{P}(A_i \mid x_t, \widetilde{g}_t < g_t^A)} \ . \tag{4.30}$$

Recall that $A_1 = \{g_t \geq g_t^A\}$. Since by definition $g_t \leq \widetilde{g}_t$, conditionally on $\{\widetilde{g}_t < g_t^A\}$, the probability of A_1 is zero. Consequently the above reduces to

$$\begin{aligned} P &= \frac{\sum_{i=2}^3 \mathcal{P}(g_t = \widetilde{g}_t \mid x_t, A_i, \widetilde{g}_t < g_t^A)\mathcal{P}(A_i \mid x_t, \widetilde{g}_t < g_t^A)}{\sum_{i=2}^3 \mathcal{P}(A_i \mid x_t, \widetilde{g}_t < g_t^A)} \\ &\geq \frac{p}{|\mathcal{A}|}\frac{\sum_{i=2}^3 \mathcal{P}(A_i \mid x_t, \widetilde{g}_t < g_t^A)}{\sum_{i=2}^3 \mathcal{P}(A_i \mid x_t, \widetilde{g}_t \leq g_t^A)} = \frac{p}{|\mathcal{A}|} \ . \end{aligned} \tag{4.31}$$

Where the last inequality is because for $i = 2, 3$ we have $\mathcal{P}(g_t = \widetilde{g}_t \mid x_t, A_i, \widetilde{g}_t < g_t^A) \geq \frac{p}{|\mathcal{A}|}$. Indeed for A_3 (case A_2 is similar) denote

$$\begin{aligned} P_1 &= \mathcal{P}(g_t = \widetilde{g}_t \mid x_t, A_3, \widetilde{g}_t < g_t^A) \\ &= \mathcal{P}(\max_{s\in\mathcal{V}_t} \langle r_t, s\rangle = \max_{s\in\mathcal{A}} \langle r_t, s\rangle \mid x_t, \max_{s\in\mathcal{V}_t} \langle r_t, s\rangle < C_0, \max_{s\in\mathcal{A}} \langle r_t, s\rangle < C_0, \alpha_{v_t}^{(t)}/(1-\alpha_{v_t}^{(t)}) < 1, \gamma_t^* < \alpha_{v_t}^{(t)}/(1- \end{aligned}$$

with $C_0 \triangleq g_t^A + \langle r_t, v_t\rangle$ and $r_t = -\nabla f(x_t)$ not depending on the t^{th} sub-sampling. Also the event $\{\max_{s\in\mathcal{V}_t} \langle r_t, s\rangle < C_0\}$ is a consequence of $\{\max_{s\in\mathcal{A}} \langle r_t, s\rangle < C_0\}$ so that P_1 simplifies to

$$P_1 = \mathcal{P}(\max_{s\in\mathcal{V}_t} \langle r_t, s\rangle = \max_{s\in\mathcal{A}} \langle r_t, s\rangle \mid x_t, \max_{s\in\mathcal{A}} \langle r_t, s\rangle < C_0, \alpha_{v_t}^{(t)}/(1-\alpha_{v_t}^{(t)}) < 1, \gamma_t^* < \alpha_{v_t}^{(t)}/(1-\alpha_{v_t}^{(t)})).$$

By definition

$$\gamma_t^* \in \underset{\gamma \in [0, \frac{\alpha_{v_t}^{(t)}}{1-\alpha_{v_t}^{(t)}}]}{\operatorname{argmin}} f(x_t + \gamma d_t^A) \ ,$$

so that γ_t^* does not depend on the t^{th} sub-sampling. Finally all the conditioning in the probability P_1 do not depend on this t^{th} sub-sampling. Hence we are in the position of using Lemma 4.B.2 for the sequence $(\langle r_t, s \rangle)_{s \in \mathcal{A}}$. Also by definition of $\mathcal{V}_t = \mathcal{S}_t \cup \mathcal{A}_t$, we have $|\mathcal{V}_t| \geq p$ so that we finally get

$$\mathcal{P}(g_t = \widetilde{g}_t \mid x_t, A_3, \widetilde{g}_t < g_t^A) \geq \frac{p}{|\mathcal{A}|} \ . \tag{4.33}$$

This was what was needed to conclude (4.31).
Case $\widetilde{g}_t \geq g_t^A$. In such a case, P from (4.29) rewrites as

$$P = \frac{\sum_{i=1}^3 \mathcal{P}(g_t = \widetilde{g}_t \mid x_t, A_i, \widetilde{g}_t \geq g_t^A) \mathcal{P}(A_i \mid x_t, \widetilde{g}_t \geq g_t^A)}{\sum_{i=1}^3 \mathcal{P}(A_i \mid x_t, \widetilde{g}_t \geq g_t^A)} \ . \tag{4.34}$$

Here $\mathcal{P}(g_t = \widetilde{g}_t \mid x_t, A_i, \widetilde{g}_t \geq g_t^A) = 0$ for $i = 2, 3$ because A_i implies $g_t < g_t^A$. So that when $\widetilde{g}_t \geq g_t^A$ it is then impossible for g_t to equal $\widetilde{g}_t$.

$$P = \frac{\mathcal{P}(g_t = \widetilde{g}_t \mid x_t, A_1, \widetilde{g}_t \geq g_t^A) \mathcal{P}(A_1 \mid x_t, \widetilde{g}_t \geq g_t^A)}{\sum_{i=1}^3 \mathcal{P}(A_i \mid x_t, \widetilde{g}_t \geq g_t^A)} \ .$$

Here also we use, and prove later on (see §below the conclusion of the proof of the Lemma), the lower bound

$$\mathcal{P}(g_t = \widetilde{g}_t \mid x_t, A_1, \widetilde{g}_t \geq g_t^A) \geq \frac{p}{|\mathcal{A}|} \ , \tag{4.35}$$

that implies

$$P \geq \frac{p}{|\mathcal{A}|} \frac{\mathcal{P}(A_1 \mid x_t, \widetilde{g}_t \geq g_t^A)}{\sum_{i=1}^3 \mathcal{P}(A_i \mid x_t, \widetilde{g}_t \geq g_t^A)} \ .$$

Because the A_i are disjoint, $\sum_{i=1}^3 \mathcal{P}(A_i \mid x_t, \widetilde{g}_t \geq g_t^A) \leq 1$ we have

$$P \geq \frac{p}{|\mathcal{A}|} \mathcal{P}(A_1 \mid x_t, \widetilde{g}_t \geq g_t^A) \ .$$

Using a similar lower bound as (4.35), namely

$$\mathcal{P}(A_1 \mid x_t, \widetilde{g}_t \geq g_t^A) \geq \frac{p}{|\mathcal{A}|} \ , \tag{4.36}$$

we finally get

$$P \geq \Big(\frac{p}{|\mathcal{A}|} \Big)^2 \ . \tag{4.37}$$

Since it is hard to precisely count the occurrences of $\{\widetilde{g}_t \geq g_t^A\}$ and $\{\widetilde{g}_t < g_t^A\}$, we use a conservative bound in (4.37)

$$\mathcal{P}(g_t = \widetilde{g}_t \mid x_t, z_t = 1) \geq \Big(\frac{p}{|\mathcal{A}|}\Big)^2 . \tag{4.38}$$

This will of course make our bound on the rate of convergence very conservative.
Justification for (4.35) **and** (4.36).

Lets denote the left hand side of(4.35) by P_2. By definition of g_t and $\widetilde{g}_t$, with $r_t = -\nabla f(x_t)$, we have:

$$P_2 = \mathcal{P}(\max_{s\in\mathcal{V}_t} \langle r_t, s - v_t\rangle = \max_{s\in\mathcal{A}} \langle r_t, s - v_t\rangle \mid x_t, \max_{s\in\mathcal{V}_t} \langle r_t, s - v_t\rangle \geq g_t^A, \max_{s\in\mathcal{A}} \langle r_t, s - v_t\rangle \geq g_t^A) \tag{4.39}$$

$$= \mathcal{P}(\max_{s\in\mathcal{V}_t} \langle r_t, s\rangle = \max_{s\in\mathcal{A}} \langle r_t, s\rangle \mid x_t, \max_{s\in\mathcal{V}_t} \langle r_t, s\rangle \geq C_0, \max_{s\in\mathcal{A}} \langle r_t, s\rangle \geq C_0) , \tag{4.40}$$

where $C_0 \triangleq g_t^A + \langle r_t, v_t\rangle$ and r_t does not depend on the random sampling at iteration t. Bayes formula leads to

$$P_2 = \frac{\mathcal{P}(\{\max_{s\in\mathcal{V}_t} \langle r_t, s\rangle = \max_{s\in\mathcal{A}} \langle r_t, s\rangle\} \cap \{\max_{s\in\mathcal{V}_t} \langle r_t, s\rangle \geq C_0\} \mid x_t, \max_{s\in\mathcal{A}} \langle r_t, s\rangle \geq C_0)}{\mathcal{P}(\max_{s\in\mathcal{V}_t} \langle r_t, s\rangle \geq C_0 \mid x_t, \max_{s\in\mathcal{A}} \langle r_t, s\rangle \geq C_0)} . \tag{4.41}$$

Conditionally on $\{\max_{s\in\mathcal{A}} \langle r_t, s\rangle \geq C_0\}$, the event $\{\max_{s\in\mathcal{V}_t} \langle r_t, s\rangle = \max_{s\in\mathcal{A}} \langle r_t, s\rangle\}$ implies $\{\max_{s\in\mathcal{V}_t} \langle r_t, s\rangle \geq C_0\}$ which leads to

$$\begin{aligned} P_2 &= \frac{\mathcal{P}(\max_{s\in\mathcal{V}_t} \langle r_t, s\rangle = \max_{s\in\mathcal{A}} \langle r_t, s\rangle \mid x_t, \max_{s\in\mathcal{A}} \langle r_t, s\rangle \geq C_0)}{\mathcal{P}(\max_{s\in\mathcal{V}_t} \langle r_t, s\rangle \geq C_0 \mid x_t, \max_{s\in\mathcal{A}} \langle r_t, s\rangle \geq C_0)} \\ &\geq \mathcal{P}(\max_{s\in\mathcal{V}_t} \langle r_t, s\rangle = \max_{s\in\mathcal{A}} \langle r_t, s\rangle \mid x_t, \max_{s\in\mathcal{A}} \langle r_t, s\rangle \geq C_0) \geq \frac{p}{|\mathcal{A}|} , \end{aligned}$$

where the last inequality is a consequence of applying Lemma 2 on the sequence $(\langle r_t, s\rangle)_{s\in\mathcal{A}}$

Similarly let's denote the left hand side of (4.36) by P_3. The first inequality is justified because conditionally on $\{\widetilde{g}_t \geq g_t^A\}$, $\{g_t = \widetilde{g}_t\} \subset \{g_t \geq g_t^A\}$. The last inequality by applying, similarly as for (4.35), Lemma 4.B.2 on the sequence $(\langle r_t, s\rangle)_{s\in\mathcal{A}}$.

$$\begin{aligned} P_3 &= \mathcal{P}(g_t \geq g_t^A \mid x_t, \widetilde{g}_t \geq g_t^A) \\ &\geq \mathcal{P}(g_t = \widetilde{g}_t \mid x_t, \widetilde{g}_t \geq g_t^A), \\ &\geq \mathcal{P}(\max_{s\in\mathcal{V}_t} \langle r_t, s\rangle = \max_{s\in\mathcal{A}} \langle r_t, s\rangle \mid x_t, \max_{s\in\mathcal{A}} \langle r_t, s\rangle \geq C_0) \geq \frac{p}{|\mathcal{A}|}. \end{aligned}$$

■

4.B.2 Main proof

Theorem 4.3.1′. *Consider the set $\mathcal{C} = \mathrm{conv}(\mathcal{A})$, with $\mathcal{A}$ a finite set of extreme atoms, after T iterations of Algorithm 11 (RAFW) we have the following linear convergence rate*

$$\mathbb{E}[h(x_{T+1})] \leq (1 - \eta^2\rho_f)^{\max\{0, \lfloor (T-s)/2 \rfloor\}} h(x_0) , \tag{4.42}$$

with $\rho_f = \frac{\mu_f^A}{4C_f^A}$, $\eta = \frac{p}{|\mathcal{A}|}$ and $s = |\mathcal{S}_0|$.

Proof. The classical curvature constant used in proofs related to non-Away Frank-Wolfe is

$$C_f \coloneqq \sup_{\substack{x,s\in\mathcal{C},\gamma\in[0,1]\\ y=x+\gamma(s-v)}} \frac{2}{\gamma^2}(f(y)-f(x)-\langle\nabla f(x),y-x\rangle) \ . \tag{4.43}$$

It is tailored for algorithms in which the update is of the form $x_{t+1}=(1-\gamma)x_t+\gamma v_t$, but this is not the shape of all updates in away versions of FW. In Lacoste-Julien and Jaggi [2015b] they introduced a modification of the above curvature constant that we also use to analyze RAFW. It is defined in [Lacoste-Julien and Jaggi, 2015b, equation (26)] as

$$C_f^A \coloneqq \sup_{\substack{x,s,v\in\mathcal{C},\gamma\in[0,1]\\ y=x+\gamma(s-v)}} \frac{2}{\gamma^2}(f(y)-f(x)-\gamma\langle\nabla f(x),s-v\rangle) \ . \tag{4.44}$$

It differs from C_f (4.43) because it allows to move outside of the domain $\mathcal{C}$. We thus require L-lipschitz continuous function on any compact set for that quantity to be upper-bounded. We refer to §**curvature constants** on [Lacoste-Julien and Jaggi, 2015b, Appendix D] for thorough details. The first part of the proof reuses the scheme of [Lacoste-Julien and Jaggi, 2015b, Theorem 8].

First part. *Upper bounding* h_t: Considering an iterate x_t that is not optimal (e.g. $x_t \neq x^*$), from [Lacoste-Julien and Jaggi, 2015b, Eq. (28)], we have

$$f(x_t)-f(x^*)=h_t \leq \frac{\widetilde{g}_t^2}{2\mu_f^A} \ , \tag{4.45}$$

where $\widetilde{g}_t$ is the *pair-wise dual gap* defined by $\widetilde{g}_t=\langle\widetilde{s}_t-v_t,-\nabla f(x_t)\rangle$. $\widetilde{s}_t$ and v_t are respectively the FW atom and the away atom in the classical Away step algorithm (conditionally on x_t, the away atom of the randomized variant coincides with the away atom of the non-randomized variant). The result is still valid here as it only uses the definition of the affine invariant version of the strong convexity parameter and does not depend on the way x_t are constructed (see *upper bounding* h_t in [Lacoste-Julien and Jaggi, 2015b, Proof for AFW in Theorem 8]).

Note that this implicitly assumes the away atom to be defined, e.g. the support of the iterate x_t never to be zero. This is ensured by the algorithm simply because it always does convex updates.

Second part. *Lower bounding progress* h_t-h_{t+1}. Consider x_t a non-optimal iterate. At step t, the update in Algorithm 11 writes $x_{t+1}(\gamma)=x_t+\gamma d_t$. γ is optimized by line-search in the segment $[0,\gamma_{\max}]$. Because in either cases d_t is a difference between two elements of $\mathcal{C}$, from the definition of C_f^A and because of the exact line search, we have

$$f(x_{t+1}) \leq \min_{\gamma\in[0,\gamma_{\max}]}(f(x_t)+\gamma\langle\nabla f(x_t),d_t\rangle+\frac{\gamma^2}{2}C_f^{\mathcal{A}}) \ ,$$

so that for any $\gamma\in[0;\gamma_{\max}]$

$$f(x_{t+1})-f(x_t) \leq \gamma\langle\nabla f(x_t),d_t\rangle+\frac{\gamma^2}{2}C_f^{\mathcal{A}}$$

or again

$$\gamma\frac{g_t}{2}-\frac{\gamma^2}{2}C_f^A \leq f(x_t)-f(x_{t+1}), \tag{4.46}$$

where the last inequality is a consequence of Lemma 4.B.1. We write $\gamma_t^B \triangleq \frac{g_t}{2C_f^A} \geq 0$, the minimizer of the left hand side of (4.46).

Case $\gamma_{\max} \geq 1$ and $\gamma_t^B \leq \gamma_{\max}$. (4.46) evaluated on $\gamma = \gamma_t^B$ gives

$$\begin{aligned} \frac{g_t^2}{4C_f^A} - \frac{g_t^2}{8C_f^A} &\leq f(x_t) - f(x_{t+1}) \\ \implies (\frac{g_t}{\widetilde{g}_t})^2 \frac{\widetilde{g}_t^2}{8C_f^A} &\leq h_t - h_{t+1}. \end{aligned} \tag{4.47}$$

Indeed, x_t is assumed not to be optimal, so that $\widetilde{g}_t \neq 0$. Combining (4.47) with (4.45) gives

$$h_{t+1} \leq h_t - (\frac{g_t}{\widetilde{g}_t})^2 \frac{\widetilde{g}_t^2}{8C_f^A} \tag{4.48}$$

$$\leq h_t - (\frac{g_t}{\widetilde{g}_t})^2 \frac{\mu_f^A}{4C_f^A} h_t \tag{4.49}$$

$$= (1 - \rho_f (\frac{g_t}{\widetilde{g}_t})^2) h_t \ . \tag{4.50}$$

Case $\gamma_{\max} \geq 1$ and $\gamma_t^B > \gamma_{\max}$. $\gamma_t^B = \frac{g_t}{2C_f^A}$ implies $g_t \geq 2C_f^A$. (4.46) transforms into

$$\begin{aligned} \frac{g_t}{2}(\gamma - \frac{\gamma^2}{2}) &\leq f(x_t) - f(x_{t+1}) \\ \frac{g_t}{\widetilde{g}_t}\frac{\widetilde{g}_t}{2}(\gamma - \frac{\gamma^2}{2}) &\leq f(x_t) - f(x_{t+1}) \ . \end{aligned}$$

Using $\widetilde{g}_t \geq h_t$ and evaluating at $\gamma = 1$, leaves us with

$$h_{t+1} \leq (1 - \frac{1}{4}\frac{g_t}{\widetilde{g}_t}) h_t. \tag{4.51}$$

Because $\mu_f^A \leq C_f^A$ [Lacoste-Julien and Jaggi, 2015b, Remark 7.] and $\rho_f = \frac{\mu_f^A}{4C_f^A}$, the two previous cases resolve in the following inequality

$$h_{t+1} \leq (1 - \rho_f (\frac{g_t}{\widetilde{g}_t})^2) h_t \ . \tag{4.52}$$

Case $\gamma_{\max} < 1$ and $\gamma_t^* < \gamma_{\max}$. By definition

$$\gamma_t^* = \operatorname*{argmin}_{\gamma \in [0, \gamma_{\max}]} f(x_t + \gamma d_t) = F(\gamma) \ . \tag{4.53}$$

f is convex and its minimum on $[0; \gamma_{\max}]$ is not reached at $\gamma_{\max}$. It is then also a minimum on the interval $[0; +\infty]$, and in particular we have

$$\gamma_t^* = \operatorname*{argmin}_{\gamma \in [0,1]} f(x_t + \gamma d_t) = F(\gamma) \ . \tag{4.54}$$

(4.46) can then be written with $\gamma \in [0, 1]$ which leads to the previous case result (4.52).

Case $\gamma_{\max} < 1$ and $\gamma_t^* = \gamma_{\max}$. This corresponds to a particular drop step for which we only guarantee $h_{t+1} \leq h_t$ (exact line-search). We call this case a *bad drop step* (indeed $\gamma_{\max} > 1$ and $\gamma_t^* = \gamma_{\max}$ also corresponds to a drop step, but for which we can prove a bound of the form $h_{t+1} \leq h_t(1 - \rho_f(\frac{g_t}{\widetilde{g}_t})^2))$.

We use the binary indicator z_t to distinguish between the step where (4.52) is guaranteed or not. Denote by $z_t = 0$ when doing a *bad drop step* and $z_t = 1$ otherwise. The second part can be summed-up in

$$h_{t+1} \leq h_t(1 - \rho_f(\frac{g_t}{\widetilde{g}_t})^2)^{z_t}. \tag{4.55}$$

Last part. Consider starting RAFW (Algorithm 11) for T iterations at $x_0 \in \text{conv}(\mathcal{V})$, with $s = |\mathcal{S}_0| \geq 0$. We will now prove there are at most $\left\lfloor \frac{T+s}{2} \right\rfloor$ drop steps. Let D_T be the number of drop steps after iteration T and F_T the number of FW step adding a new atom until iteration T. By definition, a FW step is not a drop step so that $D_T + F_T \leq T$. Also $|S_T| = |S_0| + |F_T| - |D_T|$, hence $|\mathcal{S}_T| \leq |\mathcal{S}_0| - 2|D_T| + T$ so that $|D_T| \leq \frac{T+s-|\mathcal{S}_T|}{2}$. Finally because $|\mathcal{S}_T| \geq 0$, we have $|D_T| \leq \left\lfloor \frac{T+s}{2} \right\rfloor$.

From the first two parts of the main proof, we have that

$$h_T \leq h_0 \prod_{t=0}^{T-1} (1 - \rho_f(\frac{g_t}{\widetilde{g}_t})^2)^{z_t}, \tag{4.56}$$

where $(g_t, z_t)_{t \in [0:T-1]}$ are defined along RAFW starting at x_0. For $i < j$, we write $\mathbb{E}_{i:j}$ the expectation with respect to all sub-sampling between the i^{th} iteration and the j^{th} iteration included. When taking expectation only over sub-sampling i, we write it $\mathbb{E}_i$.

We will now prove by recurrence on $T \in \mathbb{N}^*$ that

$$\mathbb{E}_{0:T-1}(\prod_{t=0}^{T-1} (1 - \rho_f(\frac{g_t}{\widetilde{g}_t})^2)^{z_t}) \leq (1 - \rho_f \eta^2)^{\max\{0, T - \lfloor \frac{T+s}{2} \rfloor\}} = F(T, s) \quad \forall s \in \mathbb{N} \ \ \forall x_0 \in \mathbb{R}^d \ \text{ with } \ |\mathcal{S}_0| = s \tag{4.57}$$

where $x_0 = \sum_{v \in \mathcal{A}} \alpha_v^{(0)} v$ and $\mathcal{S}_0 = \{v \in \mathcal{A} \text{ s.t. } \alpha_v^{(0)} > 0\}$.

The rate quantity $\max\{0, T - \left\lfloor \frac{T+s}{2} \right\rfloor\}$ represents the number of steps (between iteration 0 and $T-1$) in which $z_t = 1$, e.g. the steps in which there is a possibility of having geometrical decrease. Note that the geometrical decrease happens only when $g_t = \widetilde{g}_t$.

The key insight in the global bound is to recall (from section 4.3) that if the support is a singleton, i.e. $|\mathcal{S}_t| = 1$, RAFW does a FW step hence $z_t = 1$. We consequently distinguish whether or not the first iterate has an initial support of size 1. We then use the recurrence property starting the algorithm at x_1 and running $T - 1$ iterations.

Initialization. We will now prove the recurrence property (4.57) for $T = 1$. If $s \geq 2$, $\max\{0, T - \left\lfloor \frac{T+s}{2} \right\rfloor\} = 0$ and (4.57) is true because $(1 - \rho_f(\frac{g_0}{\widetilde{g}_0})^2) \leq 1$. If $s = 1$, this implies that the first step needs to be a Frank-Wolfe step. We necessarily have $z_0 = 1$ and so

$$\begin{aligned}
\mathbb{E}_0((1 - \rho_f(\frac{g_0}{\widetilde{g}_0})^2)^{z_0}) &= \mathbb{E}_0((1 - \rho_f(\frac{g_0}{\widetilde{g}_0})^2) \ | \ z_0 = 1) && (4.58)\\
&\leq 1 - \rho_f \mathcal{P}(g_0 = \widetilde{g}_0 \ | \ z_0 = 1) && (4.59)\\
&\leq 1 - \rho_f \eta^2 \leq 1 \leq F(1, 1) \ , && (4.60)
\end{aligned}$$

with $\eta = \frac{p}{|\mathcal{A}|}$ where F is defined in (4.57) and where the last inequality follows from (PROB) in Lemma 4.B.3.

Recurrence. Consider the property (4.57) when running $T-1$ iteration. By the tower property of conditional expectations

$$\mathbb{E}_{0:T-1}\Big(\prod_{t=0}^{T-1}\big(1-\rho_f\big(\frac{g_t}{\widetilde{g}_t}\big)^2\big)^{z_t}\Big) \;=\; \mathbb{E}_{0:T-1}\Big[\big(1-\rho_f\big(\frac{g_0}{\widetilde{g}_0}\big)^2\big)^{z_0}\mathbb{E}_{1:T-1}\Big(\prod_{t=1}^{T-1}\big(1-\rho_f\big(\frac{g_t}{\widetilde{g}_t}\big)^2\big)^{z_t}\Big)\Big] \tag{4.61}$$

We can apply the recurrence property with $T-1$ iterations and starting point x_1 on $\mathbb{E}_{1:T-1}(\prod_{t=1}^{T-1}\big(1-\rho_f\big(\frac{g_t}{g_t}\big)^2\big)^{z_t})$ so that

$$\mathbb{E}_{0:T-1}\Big(\prod_{t=0}^{T-1}\big(1-\rho_f\big(\frac{g_t}{\widetilde{g}_t}\big)^2\big)^{z_t}\Big) \;\leq\; \mathbb{E}_0\Big[\big(1-\rho_f\big(\frac{g_0}{\widetilde{g}_0}\big)^2\big)^{z_0}F(T-1,|\mathcal{S}_1|)\Big]\;, \tag{4.62}$$

where $|\mathcal{S}_1|$, the support of x_1, depends on z_0. Indeed $z_0=0$ implies a drop step and as such it decreases the support of the iterate. Thus we have to distinguish the case according to the size of the support of x_0.

Case $|\mathcal{S}_0|=1$. With $x_0=0$, RAFW starts with a FW step and as such $z_0=1$ as well as $2\geq|\mathcal{S}_1|\geq 1$ so that

$$\begin{aligned}\mathbb{E}_{0:T-1}\Big(\prod_{t=0}^{T-1}\big(1-\rho_f\big(\frac{g_t}{\widetilde{g}_t}\big)^2\big)^{z_t}\Big) &= \mathbb{E}_0\Big[\big(1-\rho_f\big(\frac{g_0}{\widetilde{g}_0}\big)^2\big)\;|\;z_0=1\Big]F(T-1,|\mathcal{S}_1|) &(4.63)\\ &\leq (1-\rho_f\eta^2)F(T-1,2)\leq F(T,1)\;, &(4.64)\end{aligned}$$

by applying (PROB) in Lemma 4.B.3. The last equality concludes the heredity in that case.

Case $|\mathcal{S}_0|\geq 2$. Here it is possible for z_0 to equal 0 or 1. If $z_0=1$, then $|\mathcal{S}_1|\leq|\mathcal{S}_0|+1$, while if $z_0=0$, it implies a drop step, we have $|\mathcal{S}_1|=|\mathcal{S}_0|-1$. If we decompose the expectation according to the value of z_0 we obtain

$$\begin{aligned}\mathbb{E}_{0:T-1}\Big(\prod_{t=0}^{T-1}\big(1-\rho_f\big(\frac{g_t}{\widetilde{g}_t}\big)^2\big)^{z_t}\Big) &\leq \mathcal{P}(z_0=1)\mathbb{E}_0\Big[\big(1-\rho_f\big(\frac{g_0}{\widetilde{g}_0}\big)^2\big)\;|\;z_0=1\Big]F(T-1,|\mathcal{S}_1|) &(4.65)\\ &\quad+\mathcal{P}(z_0=0)F(T-1,|\mathcal{S}_0|-1) &(4.66)\\ &\leq \mathcal{P}(z_0=1)(1-\rho_f\eta^2)F(T-1,|\mathcal{S}_0|+1)+\mathcal{P}(z_0=0)F(T-1,|\mathcal{S}_0|-1) &(4.67)\\ &\leq \mathcal{P}(z_0=1)(1-\rho_f\eta^2)F(T-1,s+1)+\mathcal{P}(z_0=0)F(T-1,s-1) &(4.68)\end{aligned}$$

We used the fact that $F(T,|\mathcal{S}_1|)\leq F(T-1,|\mathcal{S}_0|+1)$. Since we do not have access to the values of $\mathcal{P}(z_0=0)$ and $\mathcal{P}(z_0=1)$, we bound it in the following manner

$$\mathbb{E}_{0:T-1}\Big(\prod_{t=0}^{T-1}\big(1-\rho_f\big(\frac{g_t}{\widetilde{g}_t}\big)^2\big)^{z_t}\Big) \;\leq\; \max\big((1-\rho_f\eta^2)F(T-1,s+1),F(T-1,s-1)\big)\leq F(T,s)\;, \tag{4.69}$$

where the last inequality is just about writing the definition of F. It concludes the heredity result.

Conclusion: Starting RAFW at x_0, after T iterations, we have

$$h_T\leq h_0\prod_{t=0}^{T-1}\big(1-\rho_f\big(\frac{g_t}{\widetilde{g}_t}\big)^2\big)^{z_t}. \tag{4.70}$$

Applying (4.57) we get

$$\begin{aligned} \mathbb{E}_{0:T-1}(h_T) &\leq h_0(1-\rho_f\eta^2)^{\max\{0,T-\lfloor\frac{T+s}{2}\rfloor\}} \\ &\leq h_0(1-\rho_f\eta^2)^{\max\{0,\lfloor\frac{T-s}{2}\rfloor\}}. \end{aligned} \tag{4.71}$$

■

Generalized strongly convex.

Theorem 4.3.2′. *Suppose f has bounded smoothness constant C_f^A and is $\tilde{\mu}$-generally-strongly convex. Consider the set $\mathcal{C} = conv(\mathcal{A})$, with $\mathcal{A}$ a finite set of extreme atoms. Then after T iterations of Algorithm 11, with $s = |\mathcal{S}_0|$ and a p parameter of sub-sampling, we have*

$$\mathbb{E}[h(x_{T+1})] \leq (1-\eta^2\tilde{\rho}_f)^{\max\{0,\lfloor\frac{T-s}{2}\rfloor\}} h(x_0), \tag{4.72}$$

with $\tilde{\rho}_f = \frac{\tilde{\mu}}{4C_f^A}$ and $\eta = \frac{p}{|\mathcal{A}|}$.

Proof. The conclusion of proof of [Lacoste-Julien and Jaggi, 2015b, Th. 11] is that we have similarly as equation (4.45) by:

$$f(x_t) - f(x^*) = h_t \leq \frac{g_t^2}{2\tilde{\mu}_f}\ , \tag{4.73}$$

where $\tilde{\mu}_f > 0$ is a similar measure of the affine invariant strong convexity constant but for generalized strongly convex function.

We can thus write the twin of equation (4.55)

$$h_{t+1} \leq h_t(1-\tilde{\rho}_f(\frac{g_t}{\widetilde{g}_t})^2)^{z_t}\ , \tag{4.74}$$

with $\tilde{\rho}_f = \frac{\tilde{\mu}_f}{4C_f^A}$. The rest of the proof follows is the same as that of Theorem 4.3.1. ■

Chapter 5

Approximate Carathéodory using Bernstein-(Sterfling) Bounds

This last chapter steps slightly aside from the analysis or designing of Frank-Wolfe algorithms. Here, we sought to improve results for the approximate Shapley-Folkmann theorem [d'Aspremont and Colin, 2017] which relies on the application of specific versions of the approximate Carathéodory lemma.

Carathéodory's theorem states that if a point $\boldsymbol{x}$ lies in the convex hull of a set $C \subset \mathbb{R}^d$, then it can be represented as a convex combination of at most $d+1$ points in C. Approximate versions of this theorem seek to approach $\boldsymbol{x}$ using a smaller number of points, while minimizing approximation error. Error bounds in this case are typically obtained using a probabilistic argument, depend on the diameter of C, and implicitly assume that the number k of points in the decomposition is much smaller than d. Here, we present several approximate Carathéodory theorems on a polytope $\mathbf{Co}(V)$ where $V \subset \mathbb{R}^d$ is a finite set of points. We focus on regimes where the sampling ratio is close to 1, i.e. k is close to d. Our results also better capture the structure of $\mathbf{Co}(V)$, using both a diameter and a variance-like measure on V, in a Banach $(\mathbb{R}^d, ||\cdot||)$. The proofs rely on martingale concentration inequalities for sampling without replacement. In particular we extend the recent work of [Schneider, 2016] and derive a Bennett-Serfling concentration inequality on smooth Banach spaces.

Contents

5.1 Introduction to Carathéodory Lemma

Carathéodory's theorem states that if a point $\boldsymbol{x}$ lies in the convex hull of a set $C \subset \mathbb{R}^d$, then it can be represented as a convex combination of at most $d+1$ points in C. Approximate versions of this theorem seek to approach $\boldsymbol{x}$ using a smaller number of points, while minimizing approximation error. Recent results in this vein [Donahue et al., 1997a, Vershynin, 2012, Dai et al., 2014] have focused on producing tight approximation bounds and the following theorem states, for instance, an upper bound on the number of elements needed to achieve a given level of precision in ℓ_p norm, given a bound on the diameter of the set C.

Theorem 5.1.1 (Approximate Carathéodory)**.** *Let V be a finite subset of $\mathbb{R}^d$, $\boldsymbol{x} \in \mathbf{Co}(V)$ and $\varepsilon > 0$. We assume that V is bounded and we write*

$$D_p \triangleq \sup_{v \in V} \|v\|_p$$

with $p \geq 2$. Then, there exists some $m \leq 8pD_p^2/\varepsilon^2$ such that

$$\left\| \boldsymbol{x} - \sum_{i=1}^{m} \lambda_i \boldsymbol{v}_i \right\|_p \leq \varepsilon,$$

for some $\boldsymbol{v}_i \in V$ and $\lambda_i > 0$ such that $\mathbf{1}^\top \lambda = 1$.

This result is a direct consequence of Maurey's lemma [Pisier, 1981] and is based on a probabilistic argument which samples vectors $\boldsymbol{v}_i$ with replacement, using concentration inequalities to control the approximation error. It can also be seen as a direct application of a Frank-Wolfe algorithm [Frank and Wolfe, 1956] to the optimization problem

$$\underset{v \in \mathbf{Co}(V)}{\text{minimize}} \quad \|\boldsymbol{x} - \boldsymbol{v}\|_2^2, \tag{5.1}$$

where each iteration adds at most one extreme point in the representation [Clarkson, 2010a]. In the same vein, [Blum et al., 2016, Remark 2.7] notes that Theorem 5.1.1 for $p = 2$ follows from the analysis of the perceptron algorithm in [Novikoff, 1963].

Approximate Carathéodory. These types of results appear in functional analysis as a classical consequence of Maurey's lemma in e.g. [Pisier, 1981, Carl, 1985, Bourgain et al., 1989]. Donahue et al. [1997b], in particular, study the rates of convex approximation in functional spaces. See also [Bourgain et al., 2015, Lemma 31] for a very short proof. Mirrokni et al. [2015] prove that Theorem 5.1.1 is tight for $p \geq 2$ and suggest algorithmic applications to submodular minimization, while Adiprasito et al. [2018] focus on colourful versions of the approximate Carathéodory Theorem.

Approximating convex combinations via sampling is ubiquituous in many fields and results similar to approximate Carathéodory appear under many forms and names. For instance, [Althöfer, 1994] shows a version of Theorem 5.1.1 adapted to case where $p = \infty$, with applications to matrix games. More recently, [Barman, 2014] also used Theorem 5.1.1 to compute approximate Nash equilibria and solve densest bipartite subgraph problems.

These versions do not consider the case where m the number of vectors composing the convex approximation is close to N, the number of vectors of V in initial convex combination. Furthermore, these error bounds only use the diameter of $\mathbf{Co}(V)$, hence are somewhat oblivious to any other kind of structure in this set. These are the main limitations we seek to remedy in this work.

Sterfling concentration inequalities Probabilistic proofs of Theorem 5.1.1 rely on a concentration inequality which upper bounds deviations from the original convex combination. The quantities that control this upper bound establish a crucial link between the structure of V and approximation quality.

For example, *Hoeffding* bounds write approximation quality as a function of the diameter of V while *Bennett* or *Bernstein* bounds write it as functions of both the diameter and variance of V. *Serfling* bounds incorporate the influence of the sampling ratio, i.e. the number of nonzero coefficients in the convex approximation divided by the number of nonzero coefficients in the initial convex decomposition.

In particular, Serfling [1974] derives a Hoeffding-Serfling concentration inequality for real-valued random variable. Bardenet et al. [2015] extended this result to produce a Bernstein-Serfling inequality in the same context. Finally, Schneider [2016] shows an Hoeffding-Serfling bound on smooth Banach spaces, relying on martingale concentration inequalities of Pinelis [1994].

Contribution Our contribution here is twofold. First, we produce a version of Approximate Carathéodory with high sampling ratio in smooth Banach spaces. The proofs rely on a classical sampling argument but we use a Hoeffding-Serfling concentration inequality and sampling without replacement to account for the high sampling ratio.

Second, we prove a Bennett-Serfling concentration inequality on smooth Banach Spaces in this context. This produces an approximation bound using both a diameter and a variance term. The Banach space setting gives us more flexibility in computing of these quantities.

5.2 Approximate Caratheodory via Sampling

We now recall several key results extending Theorem 5.1.1 in our context.

5.2.1 High-Sampling ratio

We now focus on the scenario where the number of terms m in the approximation is close to N, *i.e.* when the sampling ratio is high. The classical proof of Theorem 5.1.1 relies on sampling with replacement which does not provide precise enough bounds. We will use results from [Serfling, 1974] on real-valued sample sums *without replacement* to produce a more precise version of the approximate Carathéodory theorem to handle the case where a high fraction of the coefficients is sampled.

Theorem 5.2.1 (High-Sampling Ratio in l_∞)**.** *Let* $\boldsymbol{x} = \sum_{j=1}^{N} \lambda_j \boldsymbol{v}_j$ *for* $V \in \mathbb{R}^{d\times N}$ *and some* $\lambda \in \mathbb{R}^N$ *such that* $\mathbf{1}^T\lambda = 1, \lambda \geq 0$*. Let* $\varepsilon > 0$ *and write* $R = \max\{R_v, R_\lambda\}$ *where*

$$\begin{cases} R_v = \max\limits_i \|\lambda_i \boldsymbol{v}_i\|_\infty \\ R_\lambda = \max\limits_i |\lambda_i| \ . \end{cases}$$

Consider m *(with* $\gamma = 2\log((d+1)/d)$*) s.t.*

$$m \geq 1 + N\,\frac{\gamma(\sqrt{N}\,R/\varepsilon)^2}{1+\gamma(\sqrt{N}\,R/\varepsilon)^2}. \tag{5.2}$$

Then, there exists some $\hat{\boldsymbol{x}} = \sum_{j\in\mathcal{J}} \mu_j \boldsymbol{v}_j$ *with* $\mu \in \mathbb{R}^m$ *and* $\mu \geq 0$*, where* $\mathcal{J} \subset [1, N]$ *has size* m *such that*

$$\begin{cases} \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_\infty \leq \varepsilon \\ \Big|\sum_{j\in\mathcal{J}} \mu_j - 1\Big| \leq \varepsilon. \end{cases}$$

Proof. Here the argument consists in approximating $\boldsymbol{x} = \sum_{j=1}^N \lambda_j \boldsymbol{v}_j$ by convex combinations of the form $\boldsymbol{S}_m = \sum_{j\in\mathcal{J}} \frac{N}{m}\lambda_j \boldsymbol{v}_j$ with $\mathcal{J}$ a subset of $[N]$ of size m. We apply several concentration inequalies to first upper bound the probability that $||\boldsymbol{x} - \hat{\boldsymbol{x}}||_\infty \geq \varepsilon$ and then the probability that $\Big|1 - \sum_{i\in\mathcal{J}} \frac{N}{m}\lambda_j\Big| \geq \varepsilon$, and use an union bound to conclude the proof. Let

$$\boldsymbol{S}_m^{(i)} = \sum_{j\in\mathcal{J}} \lambda_j \boldsymbol{v}_j^{(i)}$$

where $\mathcal{J}$ is a random subset of $[N]$ of size m, then [Serfling, 1974, Cor 1.1] shows

$$\mathbb{P}\left(\left|\frac{N}{m}\boldsymbol{S}_m^{(i)} - \boldsymbol{x}^{(i)}\right| \geq \varepsilon\right) \leq \exp\left(\frac{-\alpha_m \varepsilon^2}{2N(1-\alpha_m)R_v^2}\right)$$

where $\alpha_m = (m-1)/N$ is the sampling ratio. Consider $\beta \in [0, 1]$. To ensure $\mathbb{P}\Big(\Big|\frac{N}{m}\boldsymbol{S}_m^{(i)} - \boldsymbol{x}^{(i)}\Big| \leq \varepsilon\Big) \geq \beta$, it is sufficient that α_m satisfies (because $R \geq R_v$)

$$\frac{\alpha_m}{1-\alpha_m} \geq 2\log(1/(1-\beta))(R\sqrt{N}/\varepsilon)^2. \tag{5.3}$$

Hence with α_m as in (5.3), for all coordinate $\mathbb{P}\Big(\Big|\frac{N}{m}\boldsymbol{S}_m^{(i)} - \boldsymbol{x}^{(i)}\Big| \leq \varepsilon\Big) \geq \beta$. A union bound yields

$$\mathbb{P}(||\boldsymbol{x} - \hat{\boldsymbol{x}}||_\infty \leq \varepsilon) \geq d\beta - (d-1).$$

An Hoeffding inequality on the coefficients gives

$$\mathbb{P}\Big(\Big|\frac{N}{m}\sum_{i\in\mathcal{J}} \lambda_i - 1\Big| \geq \varepsilon\Big) \leq \exp(-\frac{\alpha_m \varepsilon^2}{2N(1-\alpha_m)R^2}),$$

and because α_m satisfies (5.3), we have

$$\mathbb{P}\Big(\Big|\frac{N}{m}\sum_{i\in\mathcal{J}} \lambda_i - 1\Big| \geq \varepsilon\Big) \leq 1 - \beta\ .$$

Write A the event $\Big\{\Big|\Big|\boldsymbol{x} - \hat{\boldsymbol{x}}\Big|\Big|_\infty \leq \varepsilon\Big\} \bigcap \Big\{\Big|\frac{N}{m}\sum_{j\in\mathcal{J}} \lambda_j - 1\Big| \leq \varepsilon\Big\}$. An union bound gives

$$\mathbb{P}(A) \geq (d+1)\beta - d.$$

Choosing $\beta = \frac{d+1/2}{d+1} < 1$ leads to $\mathbb{P}(A) > 0$ and the desired result. ∎

Here, R (which is bounded by the diameter of the set V) is the only value accounting for the geometry of V because we use an Hoeffding type concentration inequality. Note finally that N can be bounded by $d+1$ using the classical Carathéodory theorem.

5.2.2 Banach Spaces

For completeness, we recall the definition of $(2, D)$- Banach spaces [Schneider, 2016, Definition 3] and refer to [Schneider, 2016, section 3] for more details.

Definition 5.2.2. *A Banach space $(\mathcal{B}, ||\cdot||)$ is $(2, D)$-smooth if it a Banach space and there exists $D > 0$ such that*

$$||\boldsymbol{x} + \boldsymbol{y}||^2 + ||\boldsymbol{x} - \boldsymbol{y}||^2 \leq 2||\boldsymbol{x}||^2 + 2D||\boldsymbol{y}||^2$$

for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{B}$.

A Hilbert space for instance is $(2, 1)$-smooth [Schneider, 2016, §4]. Using Banach spaces provides much more versatility and can lead to important gains in measuring the variance or the diameter.

Theorem 5.2.1 uses Hoeffding-Serfling for real-valued random variables to provide error bounds in ℓ_∞ norm, while Theorem 5.1.1 produces a bound for any norm $||\cdot||_p$ with $p \geq 2$. To add flexibility to the diameter bound $R_v = \max_i \|\lambda_i \boldsymbol{v}_i\|_\infty$, we extend Theorem 5.2.1 to arbitrary norms in $(2, D)$-smooth Banach spaces (see definition 5.2.2) using a recent result by [Schneider, 2016]. The concentration inequality they prove allows us to directly handle the sample $\boldsymbol{S}_m$ as a vector of a Banach, not component-wise as in the proof of Theorem 5.2.1.

Theorem 5.2.3 (High Sampling Ratio in Banach Spaces)**.** *Let $\boldsymbol{x} = \sum_{j=1}^N \lambda_j \boldsymbol{v}_j$ for $V \in \mathbb{R}^{d\times N}$ and some $\lambda \in \mathbb{R}^N$ such that $\mathbf{1}^T\lambda = 1, \lambda \geq 0$. Let $\varepsilon > 0$ and write $R = \max\{R_v, R_\lambda\}$ where $R_v = \max_i \|\lambda_i \boldsymbol{v}_i\|$ and $R_\lambda = \max_i |\lambda_i|$, for some norm $\|\cdot\|$ such that $(\mathbb{R}^d, \|\cdot\|)$ is $(2, D)$-smooth (Definition 5.2.2). Consider m (with $\gamma = 2\log(2/(1-\beta))$ for some $\beta \in [0, 1]$) s.t.*

$$m \geq 1 + N\frac{\gamma(\sqrt{N}\, D\, R/\varepsilon)^2}{1 + \gamma(\sqrt{N}\, D\, R/\varepsilon)^2} \tag{5.4}$$

Then, there exists some $\hat{\boldsymbol{x}} = \sum_{j\in\mathcal{J}} \mu_j \boldsymbol{v}_j$ with $\mu \in \mathbb{R}^m$ and $\mu \geq 0$, where $\mathcal{J} \subset [1, N]$ of size m such that

$$\begin{cases} \|\boldsymbol{x} - \hat{\boldsymbol{x}}\| \leq \varepsilon \\ \Big|\sum_{j\in\mathcal{J}} \mu_j - 1\Big| \leq \varepsilon. \end{cases}$$

Proof. We use [Schneider, 2016, Th. 1] instead of [Serfling, 1974, Cor 1.1] in the proof of Theorem 5.2.1. We consider $(\lambda_1\boldsymbol{v}_1, \ldots, \lambda_N\boldsymbol{v}_N)$, the N elements of the Banach space. Write $\boldsymbol{S}_m = \sum_{i\in\mathcal{J}} \lambda_i \boldsymbol{v}_i$ for $\mathcal{J}$ a random subset of $[N]$ of size m. Note that $\frac{N}{m}\boldsymbol{S}_m$ is an unbiased estimate of $\boldsymbol{x}$. [Schneider, 2016, Th. 1] hence implies

$$\mathbb{P}\Big(\Big|\Big|\frac{N}{m}\boldsymbol{S}_m - \boldsymbol{x}\Big|\Big| \geq \varepsilon\Big) \leq 2\exp\Big(-\frac{\alpha_m\varepsilon^2}{2D^2R_v^2N(1-\alpha_m)}\Big).$$

Because $R > R_v$, we can replace R_v above by R. Consider $\beta \in]0, 1[$. To ensure $\mathbb{P}\Big(\Big|\Big|\frac{N}{m}\boldsymbol{S}_m - \boldsymbol{x}\Big|\Big| \leq \varepsilon\Big) \geq \beta$, it is sufficient that α_m verifies

$$\frac{\alpha_m}{1-\alpha_m} \geq 2\log(2/(1-\beta))(DR_v\sqrt{N}/\varepsilon)^2.$$

This means imposing (with $\gamma = 2\log(2/(1-\beta)) > 0$ for $\beta \in [0,1[$)

$$\alpha_m \geq \frac{\gamma(\sqrt{N}\, R\, D/\varepsilon)^2}{1+\gamma(\sqrt{N}\, R\, D/\varepsilon)^2}. \tag{5.5}$$

Let's apply again Hoeffding-Serfling to the real-valued (λ_j) (here consider it as a random variable). We hence have

$$\mathbb{P}\left(\Big|\frac{N}{m}\sum_{j\in\mathcal{J}}\lambda_j - 1\Big| \geq \varepsilon\right) \leq 2\exp\left(\frac{-\alpha_m\varepsilon^2}{2N(1-\alpha_m)R_\lambda^2}\right).$$

Again, replace R_λ above by R. Imposing (5.5) implies that

$$2\exp\left(\frac{-\alpha_m\varepsilon^2}{2N(1-\alpha_m)R_\lambda^2}\right) \leq 2\Big(\frac{1-\beta}{2}\Big)^{D^2}.$$

Write A the event $\Big\{\Big|\Big|\frac{N}{m}\boldsymbol{S}_m - \boldsymbol{x}\Big|\Big| \leq \varepsilon\Big\} \bigcap \Big\{\Big|\frac{N}{m}\sum_{j\in\mathcal{J}}\lambda_j - 1\Big| \leq \varepsilon\Big\}$. A union bound gives that

$$\mathbb{P}(A) \geq \beta - 2\Big(\frac{1-\beta}{2}\Big)^{D^2} = f(\beta),$$

which is strictly positive for some $\beta \in]0,1[$ (for instance by the mean-value theorem since $f(0) < 0$ and $f(1) = 1$). This yields the desired result. Note that we need only to choose β such that $\mathbb{P}(A) > 0$ with $\gamma = 2\log(2/(1-\beta))$ the lowest possible. Hence the best choice of γ depends on D only. For instance in Hilbert spaces, $D = 1$ and the best choice is $\beta = 1/2$ and $\gamma = 2\log(4)$. ■

5.2.3 Low Variance

In theorem 5.2.1 and 5.2.3, all that is extracted of the set V, is its diameter measure $R_v = \max_i \|\lambda_i \boldsymbol{v}_i\|_\infty$, because the proof relies again on an Hoeffding concentration inequality.

Recent results by [Bardenet et al., 2015] provide real-valued Bernstein-Serfling type inequalities where the bound depends on both the diameter R and a standard deviation.

Proposition 5.2.4 (Real-valued Bernstein-Serfling [Bardenet et al., 2015])**.** *Let $V = \{v_1, \ldots, v_N\}$ with $v_i \in \mathbb{R}$ and $(V_1, \ldots, V_m)$ the random sample without replacement in V. Then, for all $\varepsilon > 0$ and $\delta \in [0,1]$, the following concentration inequality holds*

$$\mathbb{P}\Big(\frac{1}{m}\sum_{i=1}^{m} V_i - \bar{v} \geq \varepsilon\Big) \leq \exp\Big(-\frac{m\varepsilon^2/2}{\gamma^2 + 2R\varepsilon/3}\Big) + \delta\ ,$$

where

$$\gamma^2 = (1-\alpha_m)\sigma^2 + \alpha_m\sigma R\sqrt{-\frac{2\log(\delta)}{m-1}},$$

with $\bar{v} = \frac{1}{N}\sum_i v_i$ and

$$\begin{cases} R = max_{i,j}|v_i - v_j| \\ \sigma = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N}(v_i - \bar{v})^2. \end{cases}$$

This concentration inequality can lead, using the same proof scheme as in Theorem 5.2.1 to a version of Approximate Carathéodory with high-sampling ratio accounting for both the variance and diameter of the set, in terms of l_∞ norm.

[Schneider, 2016] extended the real-valued Hoeffding-Serfling inequality of [Bardenet et al., 2015] to smooth Banach spaces and in what follows, we will show an extension of a Bennett-Serfling inequality to smooth Banach spaces.

5.2.4 High Sampling Ratio and Low Variance

We use Bennett-Serfling inequality to get the following bound.

Lemma 5.2.5. *In the setting of Theorem 5.A.5, for any $\delta_0 \in]0,1[$ and $\epsilon_0 > 0$, if the sampling ratio α_m satisfies*

$$\alpha_m \geq \frac{2\ln(2/\delta_0)\big[2(D\sigma_m^{BS})^2 + \epsilon_0 R_v/3\big]/N}{\epsilon_0^2 + 2\ln(2/\delta_0)\big[2(D\sigma_m^{BS})^2\big]/N} \ , \tag{5.6}$$

we have

$$\mathbb{P}\Big(\Big|\Big|\frac{1}{m}\sum_{i=1}^{m} V_i - \bar{\boldsymbol{v}}\Big|\Big| \geq \epsilon_0\Big) \leq \delta_0 \ . \tag{5.7}$$

Proof. Given $\delta_0 \in]0,1[$ and $\epsilon_0 > 0$, we are looking for a sampling ratio $\alpha_m = \frac{m}{N}$ such that

$$\mathbb{P}\Big(\Big|\Big|\frac{1}{m}\sum_{i=1}^{m} V_i - \bar{\boldsymbol{v}}\Big|\Big| \geq \epsilon_0\Big) \leq \delta_0 \ .$$

With Bennett-Serfling concentration inequality, it is sufficient to find α_m such that

$$\begin{aligned} 2\exp\big(-\frac{m\epsilon^2}{2\big(2\frac{N-m}{N}(D\sigma_m^{BS})^2 + \epsilon R_v/3\big)}\big) &\leq \delta_0 \\ -\frac{N\alpha_m\epsilon^2}{2(D\sigma_m^{BS})^2(1-\alpha_m) + \epsilon R_v/3} &\leq 2\ln(\delta_0/2), \end{aligned}$$

which is equivalent to

$$\begin{aligned} \alpha_m\epsilon^2 &\geq -\frac{2}{N}\ln(\delta_0/2)\big[2(D\sigma_m^{BS})^2(1-\alpha_m) + \epsilon R_v/3\big], \\ \alpha_m &\geq -\frac{\frac{2}{N}\ln(\delta_0/2)\big[2(D\sigma_m^{BS})^2 + \epsilon R_v/3\big]}{\epsilon^2 - \frac{4}{N}\ln(\delta_0/2)(D\sigma_m^{BS})^2} \ . \end{aligned}$$

For (5.7) to be true, it is sufficient that α_m satisfies the following,

$$\alpha_m \geq \frac{2\ln(2/\delta_0)\big[2(D\sigma_m^{BS})^2 + \epsilon_0 R_v/3\big]/N}{\epsilon_0^2 + 2\ln(2/\delta_0)\big[2(D\sigma_m^{BS})^2\big]/N} \ .$$

which is the desired result. ∎

We now conclude with the following Approximate Carathéodory type result.

Theorem 5.2.6 (High Sampling Ratio and low variance in Banach)**.** *Let* $\boldsymbol{x} = \sum_{j=1}^{N} \lambda_j \boldsymbol{v}_j$ *for* $V \in \mathbb{R}^{d\times N}$ *and some* $\lambda \in \mathbb{R}^N$ *such that* $\mathbf{1}^T\lambda = 1, \lambda \geq 0$*. For some norm* $\|\cdot\|$ *such that* $(\mathbb{R}^d, \|\cdot\|)$ *is* $(2, D)$*-smooth, write*

$$\begin{cases} R = \max_i \|\lambda_i \boldsymbol{v}_i\| \\ \sigma_m^{BS} = \dfrac{1}{\sqrt{\sum_{k=1}^m \frac{1}{(N-k)^2}}} \Big|\Big| \Big(\sum_{k=1}^m \dfrac{1}{(N-k)^2} \sigma_k^2 \Big)^{1/2} \Big|\Big|_\infty, \end{cases}$$

with

$$\sigma_k = \mathbb{E}_{k-1}||V_k - \mathbb{E}_{k-1}(V_k)||^2,$$

where (V_k) *are obtained via sampling without replacement from the sequence of* N *vectors* $(\lambda_i \boldsymbol{v}_i)$*. Consider* m *(with* $\gamma = 2\log(4)$*) s.t.*

$$m \geq 1 + N \, \frac{\gamma[2(\sqrt{N}\sigma_m^{BS} D)^2 + \varepsilon R/3]}{\varepsilon^2 + \gamma[2(\sqrt{N}\sigma_m^{BS} D)^2]}. \tag{5.8}$$

Then there exists some $\hat{\boldsymbol{x}} = \sum_{j\in\mathcal{J}} \mu_j \boldsymbol{v}_j$ *with* $\mu \in \mathbb{R}^m$ *and* $\mu \geq 0$*, where* $\mathcal{J} \subset [1, N]$ *has size* m*, such that*

$$\begin{cases} \|\boldsymbol{x} - \hat{\boldsymbol{x}}\| \leq \varepsilon \\ \Big| \sum_{j\in\mathcal{J}} \mu_j - 1 \Big| \leq \varepsilon. \end{cases}$$

Proof. For clarity we omit that in fact $R = \max\{R_v, R_\lambda\}$ and $D\sigma = D\max\{\sigma_v, \sigma_\lambda\}$ where σ_v and σ_λ are as in (5.10).

Consider $\varepsilon > 0$ and $\beta \in [0, 1]$. Applying Lemma 5.2.5 with $\delta = 1 - \beta$ we have that for α_m satisfying

$$\alpha_m \geq \frac{2\ln(2/(1-\beta))\big[2(D\sigma_m^{BS})^2 + \epsilon R/3\big]/N}{\epsilon^2 + 2\ln(2/(1-\beta))\big[2(D\sigma_m^{BS})^2\big]/N} \ . \tag{5.9}$$

that

$$\mathbb{P}\Big(\Big|\Big|\frac{1}{m}\sum_{i=1}^m V_i - \boldsymbol{x}\Big|\Big| \leq \epsilon\Big) \geq \beta \ .$$

We use again Bennett-Serfling to the real-valued sequence $(\frac{N}{m}\lambda_i)_i$. Because m verify (5.9), for random sample $\mathcal{J}$ of size m we have also

$$\mathbb{P}\Big(\Big|\sum_{i\in\mathcal{J}} \frac{N}{m}\lambda_i - 1\Big| \leq \epsilon\Big) \geq \beta \ .$$

Finally an union bound gives that

$$\mathbb{P}\Big(\Big\{||\boldsymbol{x} - \hat{\boldsymbol{x}}|| \leq \varepsilon\Big\} \bigcap \Big\{\Big|\sum_{i\in\mathcal{J}} \mu_j - 1\Big| \leq \varepsilon\Big\}\Big) \geq 2\beta - 1.$$

Choosing $\beta > \frac{1}{2}$ leads to the existence of the subset $\mathcal{J}$. ∎

Appendices

5.A Martingale Proof Details

Probabilistic proofs of approximate Carathéodory rely on a concentration inequality. To prove Theorem 5.2.3 we needed such a result for sampling without replacement with a Bennett or Bernstein upper bound. In what follows, we prove a Bennett-Serfling inequality on Banach spaces (cf. Theorem 5.A.5 below). This concentration inequality allows us to rewrite the upper bound involving the quantity R in Theorem 5.2.3 using a term taking into account a variance-like measure on V. This leads to an approximate Carathéodory version for high sampling ratio and low variance (Theorem 5.2.6). Note that this result is useful in other contexts than approximate Carathéodory, such as approximate Monte Carlo Markov chain algorithms [Bardenet et al., 2015] or Kernel Embeddings [Schneider, 2016].

Consider $V = \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_N\}$, a set of N vectors in a $(2, D)$-Banach space with norm $||\cdot||$ and $V_1, \dots, V_m$, the random variables resulting from sampling without replacement. $R_v \triangleq \sup_i ||\boldsymbol{v}_i||$ is the *range* of V. We introduce a specific notion of standard deviation related to that sampling scheme as follows

$$\sigma_m^{BS} \triangleq \frac{1}{\sqrt{\sum_{k=1}^m \frac{1}{(N-k)^2}}} \Big|\Big| \Big(\sum_{k=1}^m \frac{1}{(N-k)^2} \sigma_k^2 \Big)^{1/2} \Big|\Big|_\infty, \tag{5.10}$$

where we write $||\cdot||_\infty$ for essential supremum to simplify notations and also

$$\sigma_k = \mathbb{E}_{k-1} ||V_k - \mathbb{E}_{k-1}(V_k)||^2.$$

We call σ_m^{BS} a standard deviation because it is the square-root of the essential supremum of a convex combination of the terms $\sigma_k^2 = \mathbb{E}_{k-1} ||V_k - \mathbb{E}_{k-1}(V_k)||^2$. For $k = 1$, σ_1^2 is exactly the variance of V, while when $k = N - 1$, σ_k is better related to the diameter of the set V. The difference between classical notions of variance is due to sampling without replacement. However, when the index k increases, the weights also do, thus putting more weight on diameter-like measures rather than on variance-like measures. The notation σ_m^{BS} is an acronym for Bernstein-Serfling variance as a function of m. Finally, note that for smaller values of m, σ_m^{BS} is closer to a standard deviation term.

Our goal is to upper-bound the following probability

$$\mathbb{P}\Big(\Big|\Big| \frac{1}{m} \sum_{i=1}^m V_i - \frac{1}{N} \sum_{i=1}^N \boldsymbol{v}_i \Big|\Big| \geq \epsilon \Big), \tag{5.11}$$

as a function of both σ_m^{BS} and R_v. We call this bound *Serfling* because the quality of the upper-bound depends on the sampling ratio. Schneider [2016] shows an Hoeffding-Serfling bound (i.e. not depending on σ_m^{BS}) on $(2, D)$-Banach spaces, while Bardenet et al. [2015] provide a Bernstein-Serfling bound for real-valued random variable. Here we expand the result of [Schneider, 2016] to get a Bennett-Serfling inequality in $(2, D)$-Banach spaces. The proof exploits the forward martingale in [Serfling, 1974, Bardenet et al., 2015, Schneider, 2016] associated with the sampling without replacement and uses a result from [Pinelis, 1994] to conclude.

5.A.1 Forward Martingale when Sampling without Replacement

Write $V_1, \ldots, V_m$, the random variables resulting from sampling without replacement of m elements of V. Write $\bar{\boldsymbol{v}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{v}_i$ and consider $(M_k)_{k\in\mathbb{N}}$ the following random process

$$M_k = \begin{cases} \frac{1}{N-k}\sum_{i=1}^{k}(V_i - \bar{\boldsymbol{v}}) & 1 \leq k \leq m \\ M_m & \text{for } k > m \end{cases} \tag{5.12}$$

and $M_0 = 0$. It is a standard result (when $m = N-1$) that $(M_k)_{k\in\mathbb{N}}$ defines a forward martingale [Serfling, 1974, (2.7)], [Bardenet et al., 2015, Lemma 2.1] or [Schneider, 2016, Lemma 1] w.r.t. the filtration $(\mathcal{F}_k)_{k\in\mathbb{N}}$ defined as

$$\mathcal{F}_k = \begin{cases} \sigma(V_1, \ldots, V_k) & 1 \leq k \leq m \\ \sigma(V_1, \ldots, V_m) & \text{for } k > m \ . \end{cases} \tag{5.13}$$

In fact the martingale defined in (5.12) for some m_0 is also the stopped martingale at m_0 of the martingale in (5.12) defined for $m = N-1$ (which corresponds to the martingale studied in [Schneider, 2016, Lemma 1]).

Lemma 5.A.1. *For $m \in [N-1]$, $(M_k)_{k\in\mathbb{N}}$ as defined in (5.12) is a forward martingale with respect to the filtration $(\mathcal{F}_k)_{k\in\mathbb{N}}$ in (5.13).*

Proof. For $1 \leq k \leq m$, it is exactly the same computations as in [Schneider, 2016, Lemma 1.]. By definition, for $k > m$

$$\mathbb{E}(M_k \mid \mathcal{F}_{k-1}) = \mathbb{E}(M_m \mid \mathcal{F}_m) = M_m = M_{k-1} \ .$$

■

For $k \leq m$ we also have the two following results [Schneider, 2016, (3) and (5)].

Lemma 5.A.2.

$$I_k \triangleq M_k - M_{k-1} = \frac{V_k - \mathbb{E}_{k-1}(V_k)}{N-k} \tag{5.14}$$

$$||I_k|| \leq \frac{R}{N-k}, \tag{5.15}$$

where I_k denotes the martingale's increment and R is such that $max_i||\boldsymbol{v}_i|| \leq R$.

Proof. By definition of (M_k)

$$M_k = \frac{N-k+1}{N-k} M_{k-1} + \frac{V_k - \bar{\boldsymbol{v}}}{N-k} \ ,$$

hence

$$M_k - M_{k-1} = \frac{V_k - (\bar{\boldsymbol{v}} - M_{k-1})}{N-k} \ .$$

And exactly as in [Schneider, 2016, (3)], conditionally on the event $\{V_1, \ldots, V_{k-1}\}$, V_k takes its values uniformly at random from $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N\} \setminus \{V_1, \ldots, V_{k-1}\}$ so that

$$\mathbb{E}_{k-1}(V_k) = \bar{\boldsymbol{v}} - M_{k-1},$$

which finally leads to

$$||M_k - M_{k-1}|| = \frac{||V_k - \mathbb{E}_{k-1}(V_k)||}{N-k} \leq \frac{R}{N-k}$$

and the desired result. ■

5.A.2 Bennett for Martingales in Smooth Banach Spaces

We recall a slightly modified version of [Pinelis, 1994, Theorem 3.4.]. This theorem is analogous, on martingales defined on Banach spaces, of the Bennett concentration inequality for sums of real independent random variables.

Theorem 5.A.3 (Pinelis)**.** *Suppose $(M_k)_{k\in\mathbb{N}}$ is a martingale of a $(2, D)$-smooth separable Banach space and that there exists $(a, b) \in \mathbb{R}_+^*$ such that*

$$\begin{aligned} \big|\big| \sup_k ||M_k - M_{k-1}|| \big|\big|_\infty &\leq a \\ \big|\big| \big(\sum_{j=1}^{\infty} \mathbb{E}_{j-1} ||M_j - M_{j-1}||^2 \big)^{1/2} \big|\big|_\infty &\leq b/D \ , \end{aligned}$$

then for all $\eta \geq 0$,

$$\mathbb{P}(\sup_k ||M_k|| \geq \eta) \leq 2 \exp\big(- \frac{\eta^2}{2(b^2 + \eta a/3)} \big) \ .$$

Proof. Write $P = \mathbb{P}(\sup_k ||M_k|| \geq \eta)$. In the proof of [Pinelis, 1994, theorem 3.4.], we have

$$P \leq 2 \exp\big(- \lambda\eta + \frac{\exp(\lambda a) - 1 - \lambda a}{a^2} b^2 \big) \ .$$

Besides, [Sridharan, 2002, equation (16)] gives

$$\inf_{\lambda > 0} \big[- \lambda\epsilon + (e^{-\lambda} - \lambda - 1) c^2 \big] \leq - \frac{\epsilon^2}{2(c^2 + \epsilon/3)} \ .$$

We can then rewrite the initial inequality as

$$\begin{aligned} P &\leq 2 \exp\big(- \lambda a \frac{\eta}{a} + (\exp(\lambda a) - 1 - \lambda a) \frac{b^2}{a^2} \big) \\ &\leq 2 \exp\big(- \frac{\eta^2}{2(b^2 + \eta a/3)} \big) \ . \end{aligned}$$

[Pinelis, 1994] uses the exact minimization on λ which leads to a better but much less convenient form of the Bennett concentration inequality. ■

5.A.3 Bennett-Serfling in Smooth Banach Spaces

The following lemma allows to identify the parameters (a, b) appearing in theorem 5.A.3.

Lemma 5.A.4.

$$\begin{aligned} \big|\big| \sup_k ||I_k|| \big|\big|_\infty &\leq \frac{R}{N - m} \\ \big|\big| \big(\sum_{j=1}^{\infty} \mathbb{E}_{j-1} ||I_j||^2 \big)^{1/2} \big|\big|_\infty &\leq \sigma_m^{BS} \sqrt{\frac{m}{(N - m - 1)N}} \ , \end{aligned} \tag{5.16}$$

with σ_m^{BS} as in (5.10) and $I_k = M_k - M_{k-1}$.

Proof. (5.16) directly follows from (5.15). Note that $I_k = 0$ for $k \geq m$. Because of (5.14), we have

$$\Big(\sum_{k=1}^{\infty} \mathbb{E}_{k-1}(||I_k||^2)\Big)^{\frac{1}{2}} = \Big(\sum_{k=1}^{m} \frac{1}{(N-k)^2}\sigma_k^2\Big)^{\frac{1}{2}} ,$$

with $\sigma_k^2 = \mathbb{E}_{k-1}(||V_k - \mathbb{E}_{k-1}(V_k)||^2)$. Because of (5.10), we have,

$$\Big|\Big|\Big(\sum_{k=1}^{\infty} \mathbb{E}_{k-1}(||I_k||^2)\Big)^{\frac{1}{2}}\Big|\Big|_{\infty} = \sigma_m^{BS}\sqrt{\sum_{k=1}^{m}\frac{1}{(N-k)^2}} .$$

For instance, [Serfling, 1974, Lemma 2.1.] gives

$$\begin{aligned}\sum_{k=1}^{m}\frac{1}{(N-k)^2} &= \sum_{k=N-m-1+1}^{N-1}\frac{1}{k^2} \\ &\leq \frac{m}{N(N-m-1)} .\end{aligned}$$

It leads to

$$\Big|\Big|\Big(\sum_{k=1}^{\infty} \mathbb{E}_{k-1}(||I_k||^2)\Big)^{\frac{1}{2}}\Big|\Big|_{\infty} \leq \sigma_m^{BS}\sqrt{\frac{m}{N(N-m-1)}}$$

and the desired result. ∎

We finally state our main concentration inequality.

Theorem 5.A.5 (Bennett-Serfling in Banach)**.** *Consider V a discrete set of N vectors in a $(2, D)$-Banach space and $(V_i)_{i=1,\dots,m}$ the random variables obtained by sampling without replacements m elements of V. For any $\epsilon > 0$ write $P_m(\epsilon) \triangleq \mathbb{P}\Big(\Big|\Big|\frac{1}{m}\sum_{i=1}^{m} V_i - \bar{\boldsymbol{v}}\Big|\Big| \geq \epsilon\Big)$. We have*

$$P_m(\epsilon) \leq 2\exp\Big(-\frac{m\epsilon^2}{2(2\frac{N-m}{N}(D\sigma_m^{BS})^2 + \epsilon R/3)}\Big) ,$$

with $\bar{\boldsymbol{v}} \triangleq \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{v}_i$, $R \triangleq \sup_{\boldsymbol{v}\in V}||\boldsymbol{v}||$, and

$$\sigma_m^{BS} \triangleq \frac{1}{\sqrt{\sum_{k=1}^{m}\frac{1}{(N-k)^2}}}\Big|\Big|\Big(\sum_{k=1}^{m}\frac{1}{(N-k)^2}\sigma_k^2\Big)^{1/2}\Big|\Big|_{\infty} .$$

Proof. Using Theorem 5.A.3 with the forward martingale (5.12), we have for any $\eta > 0$,

$$\mathbb{P}(\sup_i ||M_i|| \geq \eta) \leq 2\exp\big(-\frac{\eta^2}{2(b^2 + \eta a/3)}\big) ,$$

and writing $P(\eta) = \mathbb{P}\Big(\frac{1}{N-m}\Big|\Big|\sum_{i=1}^{m}(V_i - \bar{\boldsymbol{v}})\Big|\Big| \geq \eta\Big)$, we have $P(\eta) \leq \mathbb{P}(\sup_i ||M_i|| \geq \eta)$. Hence

$$P\Big(\frac{N-m}{m}\eta\Big) \leq 2\exp\big(-\frac{\eta^2}{2(b^2 + \eta a/3)}\big) .$$

Because of lemma 5.A.4, $a = \frac{R}{N-m}$ and $b = D\sigma_m^{BS}\sqrt{\frac{m}{N(N-m-1)}}$ is a good choice and leads to

$$P_m\Big(\frac{N-m}{m}\eta\Big) \leq 2\exp\big(-\frac{m\epsilon}{2(2\frac{N-m}{N}(D\sigma_m^{BS})^2 + \epsilon R_v/3)}\big) ,$$

for any $\eta > 0$ with $\epsilon = \frac{N-m}{m}\eta$. ∎

Appendix A

Reconstructing Latent Orderings by Spectral Clustering

Spectral clustering uses a graph Laplacian spectral embedding to enhance the cluster structure of some data sets. When the embedding is one dimensional, it can be used to sort the items (spectral ordering). A number of empirical results also suggests that a multidimensional Laplacian embedding enhances the latent ordering of the data, if any. This also extends to circular orderings, a case where unidimensional embeddings fail. We tackle the task of retrieving linear and circular orderings in a unifying framework, and show how a latent ordering on the data translates into a filamentary structure on the Laplacian embedding. We propose a method to recover it, illustrated with numerical experiments on synthetic data and real DNA sequencing data.

Contents

A.1 Introduction

The seriation problem seeks to recover a latent ordering from similarity information. We typically observe a matrix measuring pairwise similarity between a set of n elements and assume they have a serial structure, *i.e.* they can be ordered along a chain where the similarity between elements decreases with their distance within this chain. In practice, we observe a random permutation of this similarity matrix, where the elements are not indexed according to that latent ordering. Seriation then seeks to find that global latent ordering using only (local) pairwise similarity.

Seriation was introduced in archaeology to find the chronological order of a set of graves. Each contained artifacts, assumed to be specific to a given time period. The number of common artifacts between two graves define their similarity, resulting in a chronological ordering where two contiguous graves belong to a same time period. It also has applications in, *e.g.*, envelope reduction [Barnard et al., 1995], bioinformatics [Atkins and Middendorf, 1996, Higgs et al., 2006, Cheema et al., 2010, Jones et al., 2012] and DNA sequencing [Meidanis et al., 1998, Garriga et al., 2011, Recanati et al., 2016].

In some applications, the latent ordering is circular. For instance, in *de novo* genome assembly of bacteria, one has to reorder DNA fragments subsampled from a circular genome.

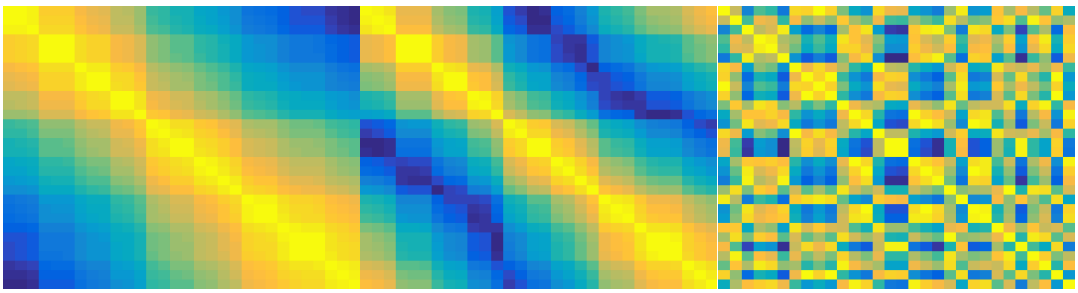
In biology, a cell evolves according to a cycle: a newborn cell passes through diverse states (growth, DNA-replication, *etc.*) before dividing itself into two newborn cells, hence closing the loop. Problems of interest then involve collecting cycle-dependent data on a population of cells at various, unknown stages of the cell-cycle, and trying to order the cells according to their cell-cycle stage. Such data include gene-expression [Liu et al., 2017], or DNA 3D conformation data [Liu et al., 2018]. In planar tomographic reconstruction, the shape of an object is inferred from projections taken at unknown angles between 0 and 2π. Reordering the angles then enables to perform the tomography [Coifman et al., 2008].

The main structural hypothesis on similarity matrices related to seriation is the concept of R-matrix, which we introduce below, together with its circular counterpart.

Definition A.1.1. *We say that $A \in \mathbf{S}_n$ is a R-matrix (or Robinson matrix) iff it is symmetric and satisfies $A_{i,j} \leq A_{i,j+1}$ and $A_{i+1,j} \leq A_{i,j}$ in the lower triangle, where $1 \leq j < i \leq n$.*

Definition A.1.2. *We say that $A \in \mathbf{S}_n$ is a circular R-matrix iff it is symmetric and satisfies, for all $i \in [n]$, $(A_{ij})_{j=1}^{i}$ and $(A_{ij})_{i=j}^{n}$ are unimodal : they are decrease to a minimum and then increase.*

Here $\mathbf{S}_n$ is the set of real symmetric matrices of dimension n. Definition A.1.1 states that when moving away from the diagonal in a given row or column of A, the entries are non-increasing, whereas in Def A.1.2, the non-increase is followed by a non-decrease. For instance, the proximity matrix of points embedded on a circle follows Def A.1.2. Figure A.1 displays examples of such matrices.



(a) R-matrix (b) circular R-matrix (c) permuted R-matrix

Figure A.1: From left to right, R-matrix, circular R-matrix, and a randomly permuted observation of a R-matrix. Seriation seeks to recover the R-matrix from its permuted observation, the permuted R-matrix.

In what follows, we write $\mathcal{L}_R^n$ (resp., $\mathcal{C}_R^n$) the set of R (resp., circular-R) matrices of size n, and $\mathcal{P}_n$ the set of permutations of n elements. A permutation can be represented by a vector π (lower case) or a matrix $\Pi \in \{0,1\}^{n\times n}$ (upper case) defined by $\Pi_{ij} = 1$ iff $\pi(i) = j$, and $\pi = \Pi\mathbf{e}$ where $\mathbf{e} = (1,\dots,n)^T$. We refer to both representations by $\mathcal{P}_n$ and may omit the subscript n whenever the dimension is clear from the context. We say that $A \in \mathbf{S}_n$ is pre-$\mathcal{L}_R$ (resp., pre-$\mathcal{C}_R$) if there exists a permutation $\Pi \in \mathcal{P}$ such that the matrix $\Pi A \Pi^T$ (whose entry (i,j) is $A_{\pi(i),\pi(j)}$) is in $\mathcal{L}_R$ (resp., $\mathcal{C}_R$). Given such A, Seriation seeks to recover this permutation Π,

$$\text{find} \quad \Pi \in \mathcal{P} \qquad \text{such that} \quad \Pi A \Pi^T \in \mathcal{L}_R \qquad \text{(Linear Seriation)}$$

$$\text{find} \quad \Pi \in \mathcal{P} \qquad \text{such that} \quad \Pi A \Pi^T \in \mathcal{C}_R \qquad \text{(Circular Seriation)}$$

A widely used method for Linear Seriation is a spectral relaxation based on the graph Laplacian of the similarity matrix. It transposes Spectral Clustering [Von Luxburg, 2007] to the case where we wish to infer a latent ordering rather than a latent clustering on the data. Roughly speaking, both methods embed the elements on a line and associate a coordinate $f_i \in \mathbb{R}$ to each element $i \in [n]$. Spectral clustering addresses a graph-cut problem by grouping these coordinates into two clusters. Spectral ordering [Atkins et al., 1998] addresses Linear Seriation by sorting the f_i.

Most Spectral Clustering algorithms actually use a Laplacian embedding of dimension $d > 1$, denoted d-**LE** in the following. Latent cluster structure is assumed to be enhanced in the d-**LE**, and the k-means algorithm [MacQueen et al., 1967, Hastie et al., 2009] seamlessly identifies the clusters from the embedding. In contrast, Spectral Ordering is restricted to $d = 1$ by the sorting step (there is no total order relation on $\mathbb{R}^d$ for $d > 1$). Still, the latent linear structure may emerge from the d-**LE**, if the points are distributed along a curve. Also, for $d = 2$, it may capture the circular structure of the data and allow for solving Circular Seriation. One must then recover a (circular) ordering of points lying in a $1D$ manifold (a curve, or filament) embedded in $\mathbb{R}^d$.

In Section A.2, we review the Spectral Ordering algorithm and the Laplacian Embedding used in Spectral Clustering. We mention graph-walk perspectives on this embedding and how this relates to dimensionality reduction techniques. Finally, we recall how these perspectives relate the discrete Laplacian to continuous Laplacian operators, providing insights about the curve structure of the Laplacian embedding through the spectrum of the limit operators. These asymptotic results were used to infer circular orderings in a tomography application in e.g. Coifman et al. [2008]. In Section A.3, we evidence the filamentary structure of the Laplacian Embedding, and provide theoretical guarantees about the Laplacian Embedding based method for Circular Seriation. We then propose a method in Section A.4 to leverage the multidimensional Laplacian embedding in the context of Linear Seriation and Circular Seriation. We eventually present numerical experiments to illustrate how the spectral method gains in robustness by using a multidimensional Laplacian embedding.

A.2 Related Work

A.2.1 Spectral Ordering for Linear Seriation

Linear Seriation can be addressed with a spectral relaxation of the following combinatorial problem,

$$\text{minimize} \quad \sum_{i,j=1}^{n} A_{ij}|\pi_i - \pi_j|^2 \quad \text{such that} \quad \pi \in \mathcal{P}_n \tag{2-SUM}$$

Intuitively, the optimal permutation compensates high A_{ij} values with small $|\pi_i - \pi_j|^2$, thus laying similar elements nearby. For any $f = (f(1), \dots, f(n))^T \in \mathbb{R}^n$, the objective of 2-SUM can be written as a quadratic (with simple algebra using the symmetry of A, see Von Luxburg [2007]),

$$\sum_{i,j=1}^{n} A_{ij}|f(i) - f(j)|^2 = f^T L_A f \tag{A.1}$$

where $L_A \triangleq \mathbf{diag}(A\mathbf{1}) - A$ is the graph-Laplacian of A. From (A.1), L_A is positive-semi-definite for A having non-negative entries, and $\mathbf{1} = (1, \dots, 1)^T$ is an eigenvector associated to $\lambda_0 = 0$.

The spectral method drops the constraint $\pi \in \mathcal{P}_n$ in 2-SUM and enforces only norm and orthogonality constraints, $\|\pi\| = 1$, $\pi^T\mathbf{1} = 0$, to avoid the trivial solutions $\pi = 0$ and $\pi \propto \mathbf{1}$, yielding,

$$\text{minimize} \quad f^T L_A f \quad \text{such that} \quad \|f\|_2 = 1\,,\ f^T\mathbf{1} = 0. \tag{Relax. 2-SUM}$$

This is an eigenvalue problem on L_A solved by $f_{(1)}$, the eigenvector associated to $\lambda_1 \geq 0$ the second smallest eigenvalue of L_A. If the graph defined by A is connected (which we assume further) then $\lambda_1 > 0$. From $f_{(1)}$, one can recover a permutation by sorting its entries. The spectral relaxation of 2-SUM is summarized in Algorithm 12. For pre-$\mathcal{L}_R$ matrices, Linear Seriation is equivalent to 2-SUM [Fogel et al., 2013], and can be solved with Algorithm 12 [Atkins et al., 1998], as stated in Theorem A.2.1.

Algorithm 12 Spectral ordering [Atkins et al., 1998]

Input: Connected similarity matrix $A \in \mathbb{R}^{n\times n}$
1: Compute Laplacian $L_A = \mathbf{diag}(A\mathbf{1}) - A$
2: Compute second smallest eigenvector of L_A, f_1
3: Sort the values of f_1
Output: Permutation $\sigma : f_1(\sigma(1)) \leq \ldots \leq f_1(\sigma(n))$

Theorem A.2.1 (Atkins et al. [1998]). *If $A \in \mathbf{S}_n$ is a pre-$\mathcal{L}_R$ matrix, then Algorithm 12 recovers a permutation $\Pi \in \mathcal{P}_n$ such that $\Pi A \Pi^T \in \mathcal{L}_R^n$, i.e., it solves Linear Seriation.*

A.2.2 Laplacian Embedding

Let $0 = \lambda_0 < \lambda_1 \leq \ldots \leq \lambda_{n-1}$, $\Lambda \triangleq \mathbf{diag}\,(\lambda_0, \ldots, \lambda_{n-1})$, $\Phi = (\mathbf{1}, f_1, \ldots, f_{n-1})$, be the eigendecomposition of $L_A = \Phi \Lambda \Phi^T$. Algorithm 12 embeds the data in 1D through the eigenvector f_1 (1-**LE**). For any $d < n$, $\Phi^{(d)} \triangleq (f_1, \ldots, f_d)$ defines a d-dimensional embedding (d-**LE**)

$$\boldsymbol{y}_i = (f_1(i), f_2(i), \ldots, f_d(i))^T \in \mathbb{R}^d, \quad \text{for} \quad i = 1, \ldots, n. \tag{d-\textbf{LE}}$$

which solves the following embedding problem,

$$\begin{array}{ll} \text{minimize} & \sum_{i,j=1}^n A_{ij} \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2 \\ \text{such that} & \tilde{\Phi} = \left(\boldsymbol{y}_1^T, \ldots, \boldsymbol{y}_n^T\right)^T \in \mathbb{R}^{n \times d}\,,\ \tilde{\Phi}^T \tilde{\Phi} = \mathbf{I}_d\,,\ \tilde{\Phi}^T \mathbf{1}_n = \mathbf{0}_d \end{array} \tag{Lap-Emb}$$

Indeed, like in (A.1), the objective of Lap-Emb can be written $\mathbf{Tr}\left(\tilde{\Phi}^T L_A \tilde{\Phi}\right)$ (see Belkin and Niyogi [2003] for a similar derivation). The 2-SUM intuition still holds: the d-**LE** lays similar elements nearby, and dissimilar apart, in $\mathbb{R}^d$. Other dimensionality reduction techniques such as Multidimensional scaling (MDS) [Kruskal and Wish, 1978], kernel PCA [Schölkopf et al., 1997], or Locally Linear Embedding (LLE) [Roweis and Saul, 2000] could be used as alternatives to embed the data in a way that intuitively preserves the latent ordering. However, guided by the generalization of Algorithm 12 and theoretical results that follow, we restrict ourselves to the Laplacian embedding.

Normalization and Scaling

Given the weighted adjacency matrix $W \in \mathbf{S}_n$ of a graph, its Laplacian reads $L = D - W$, where $D = \mathbf{diag}(W\mathbf{1})$ has diagonal entries $d_i = \sum_{j=1}^n W_{ij}$ (degree of i). Normalizing W_{ij} by $\sqrt{d_i d_j}$ or d_i leads to the normalized Laplacians,

$$\begin{array}{rcl} L^{\text{sym}} & = & D^{-1/2} L D^{-1/2} = \mathbf{I} - D^{-1/2} W D^{-1/2} \\ L^{\text{rw}} & = & D^{-1} L = \mathbf{I} - D^{-1} W \end{array} \tag{A.2}$$

They correspond to graph-cut normalization (normalized cut or ratio cut). Moreover, L^{rw} has a Markov chain interpretation, where a random walker on edge i jumps to edge j from time t to $t+1$ with transition probability $P_{ij} \triangleq W_{ij}/d_i$. It has connections with diffusion processes, governed by the heat equation $\frac{\partial \mathcal{H}_t}{\partial t} = -\Delta \mathcal{H}_t$, where Δ is the Laplacian operator, $\mathcal{H}_t$ the heat kernel, and t is time [Qiu and Hancock, 2007]. These connections lead to diverse Laplacian embeddings backed by theoretical justifications, where the eigenvectors f_k^{rw} of L^{rw} are sometimes scaled by decaying weights α_k (thus emphasizing the first eigenvectors),

$$\tilde{\boldsymbol{y}}_i = (\alpha_1 f_1^{\text{rw}}(i), \ldots, \alpha_{d-1} f_d^{\text{rw}}(i))^T \in \mathbb{R}^d, \quad \text{for} \quad i = 1, \ldots, n. \tag{(α, d)-\textbf{LE}}$$

Laplacian eigenmaps [Belkin and Niyogi, 2003] is a nonlinear dimensionality reduction technique based on the spectral embedding of L^{rw} ((α, d)-**LE**) with $\alpha_k = 1$ for all k). Specifically, given points $x_1, \ldots, x_n \in \mathbb{R}^d$, the method computes a heat kernel similarity matrix $W_{ij} = \exp - (\|x_i - x_j\|^2/t)$ and outputs the first eigenvectors of L^{rw} as a lower dimensional

embedding. The choice of the heat kernel is motivated by connections with the heat diffusion process on a manifold, a partial differential equation involving the Laplacian operator. This method has been successful in many machine learning applications such as semi-supervised classification [Belkin and Niyogi, 2004] and search-engine type ranking [Zhou et al., 2004]. Notably, it provides a global, nonlinear embedding of the points that preserves the local structure.

The commute time distance $\mathrm{CTD}(i,j)$ between two nodes i and j on the graph is the expected time for a random walker to travel from node i to node j and then return. The full (α, d)-**LE**, with $\alpha_k = (\lambda_k^{\mathrm{rw}})^{-1/2}$ and $d = n-1$, satisfies $\mathrm{CTD}(i,j) \propto \|\tilde{\boldsymbol{y}}_i - \tilde{\boldsymbol{y}}_j\|$. Given the decay of α_k, the d-**LE** with $d \ll n$ approximately preserves the CTD. This embedding has been successfully applied to vision tasks, *e.g.*, anomaly detection [Albano and Messinger, 2012], image segmentation and motion tracking [Qiu and Hancock, 2007].

Another, closely related dimensionality reduction technique is that of diffusion maps [Coifman and Lafon, 2006], where the embedding is derived to preserve diffusion distances, resulting in the (α, d)-**LE**, for $t \geq 0$, $\alpha_k(t) = (1-\lambda_k^{\mathrm{rw}})^t$.

Coifman and Lafon [2006], Coifman et al. [2008] also propose a normalization of the similarity matrix $\tilde{W} \leftarrow D^{-1}WD^{-1}$, to extend the convergence of L^{rw} towards the Laplace-Beltrami operator on a curve when the similarity is obtained through a heat kernel on points that are *non uniformly* sampled along that curve.

Finally, we will use in practice the heuristic scaling $\alpha_k = 1/\sqrt{k}$ to damp high dimensions. For a deeper discussion about spectral graph theory and the relations between these methods, see for instance Qiu and Hancock [2007] and Chung and Yau [2000].

A.2.3 Link with Continuous Operators

In the context of dimensionality reduction, when the data points $x_1, \ldots, x_n \in \mathbb{R}^D$ lie on a manifold $\mathcal{M} \subset \mathbb{R}^d$ of dimension $K \ll D$, the graph Laplacian L of the heat kernel ($W_{ij} = \exp\left(-\|x_i - x_j\|^2/t\right)$) used in Belkin and Niyogi [2003] is a discrete approximation of $\Delta_{\mathcal{M}}$, the Laplace-Beltrami operator on $\mathcal{M}$ (a differential operator akin to the Laplace operator, adapted to the local geometry of $\mathcal{M}$). Singer [2006] specify the hypothesis on the data and the rate of convergence of L towards $\Delta_{\mathcal{M}}$ when n grows and the heat-kernel bandwidth t shrinks. Von Luxburg et al. [2005] also explore the spectral asymptotics of the spectrum of L to prove consistency of spectral clustering.

This connection with continuous operators gives hints about the Laplacian embedding in some settings of interest for Linear Seriation and Circular Seriation. Indeed, consider n points distributed along a curve $\Gamma \subset \mathbb{R}^D$ of length 1, parameterized by a smooth function $\gamma : \mathbb{R} \to \mathbb{R}^D$, $\Gamma = \{\gamma(s) \,:\, s \in [0,1]\}$, say $x_i = \gamma(i/n)$. If their similarity measures their proximity along the curve, then the similarity matrix is a circular-R matrix if the curve is closed ($\gamma(0) = \gamma(1)$), and a R matrix otherwise. Coifman et al. [2008] motivate a method for Circular Seriation with the spectrum of the Laplace-Beltrami operator Δ_Γ on Γ when Γ is a closed curve. Indeed, Δ_Γ is simply the second order derivative with respect to the arc-length s, $\Delta_\Gamma f(s) = f''(s)$ (for f twice continuously differentiable), and its eigenfunctions are given by,

$$f''(s) = -\lambda f(s). \tag{A.3}$$

With periodic boundary conditions, $f(0) = f(1)$, $f'(0) = f'(1)$, and smoothness assumptions, the first eigenfunction is constant with eigenvalue $\lambda_0 = 0$, and the remaining are $\{\cos(2\pi m s), \sin(2\pi m s)\}_{m=1}^{\infty}$, associated to the eigenvalues $\lambda_m = (2\pi m)^2$ of multiplicity 2.

Hence, the 2-**LE**, $(f_1(i), f_2(i)) \approx (\cos(2\pi s_i), \sin(2\pi s_i))$ should approximately lay the points on a circle, allowing for solving Circular Seriation [Coifman et al., 2008]. More generally, the 2d-**LE**, $(f_1(i), \ldots, f_{2d+1}(i))^T \approx (\cos(2\pi s_i), \sin(2\pi s_i), \ldots, \cos(2d\pi s_i), \sin(2d\pi s_i))$ is a closed curve in $\mathbb{R}^{2d}$.

If Γ is not closed, we can also find its eigenfunctions. For instance, with Neumann boundary conditions (vanishing normal derivative), say, $f(0) = 1$, $f(1) = 0$, $f'(0) = f'(1) = 0$, the non-trivial eigenfunctions of Δ_Γ are $\{\cos(\pi m s)\}_{m=1}^{\infty}$, with associated eigenvalues $\lambda_m = (\pi m)^2$ of multiplicity 1. The 1-**LE** $f_1(i) \approx \cos(\pi s_i)$ respects the monotonicity of i, which is consistent with Theorem A.2.1. Lafon [2004] invoked this asymptotic argument to solve an instance of Linear Seriation but seemed unaware of the existence of Atkin's Algorithm 12. Note that here too, the d-**LE**, $(f_1(i), \ldots, f_d(i))^T \approx (\cos(\pi s_i), \ldots, \cos(d\pi s_i))$ follows a closed curve in $\mathbb{R}^d$, with endpoints.

These asymptotic results hint that the Laplacian embedding preserves the latent ordering of data points lying on a curve embedded in $\mathbb{R}^D$. However, these results are only asymptotic and there is no known guarantee for the Circular Seriation problem as there is for Linear Seriation. Also, the curve (sometimes called filamentary structure) stemming from the Laplacian embedding has been observed in more general cases where no hypothesis on a latent representation of the data is made, and the input similarity matrix is taken as is (see, *e.g.*, Diaconis et al. [2008] for a discussion about the horseshoe phenomenon).

A.2.4 Ordering Points Lying on a Curve

Finding the latent ordering of some points lying on (or close to) a curve can also be viewed as an instance of the travelling salesman problem (TSP), for which a plethora of (heuristic or approximation) algorithms exist [Reinelt, 1994, Laporte, 1992]. We can think of this setting as one where the cities to be visited by the salesman are already placed along a single road, thus these TSP instances are easy and may be solved by simple heuristic algorithms.

Existing approaches for Linear Seriation and Circular Seriation have only used 2D embeddings so far, for simplicity. Kuntz et al. [2001] use the 2-**LE** to find a circular ordering of the data. They use a somehow exotic TSP heuristic which maps the 2D points onto a pre-defined "space-filling" curve, and unroll the curve through its closed form inverse to obtain a 1D embedding and sort the points. Friendly [2002] uses the angle between the first two coordinates of the 2D-MDS embedding and sorts them to perform Linear Seriation. Coifman et al. [2008] use the 2-**LE** to perform Circular Seriation in a tomographic reconstruction setting, and use a simple algorithm that sorts the inverse tangent of the angle between the two components to reorder the points. Liu et al. [2018] use a similar approach to solve Circular Seriation in a cell-cycle related problem, but with the 2D embedding given by MDS.

A.3 Spectral properties of some (Circular) Robinson Matrices

We have claimed that the d-**LE** enhances the latent ordering of the data and we now present some theoretical evidences. We adopt a point of view similar to Atkins et al. [1998], where the feasibility of Linear Seriation relies on structural assumptions on the similarity matrix ($\mathcal{L}_R$). For a subclass $\mathcal{C}_R^*$ of $\mathcal{C}_R$ (set of circular-R matrices), we show that the d-**LE** lays the points on a closed curve, and that for $d = 2$, the elements are embedded on a circle according to their latent circular ordering. This is a counterpart of Theorem A.2.1 for Circular Seriation.

It extends the asymptotic results motivating the approach of Coifman et al. [2008], shifting the structural assumptions on the elements (data points lying on a curve embedded in $\mathbb{R}^D$) to assumptions on the raw similarity matrix that can be verified in practice. Then, we develop a perturbation analysis to bound the deformation of the embedding when the input matrix is in $\mathcal{C}_R^*$ up to a perturbation. Finally, we discuss the spectral properties of some (non circular) $\mathcal{L}_R$-matrices that shed light on the filamentary structure of their d-**LE** for $d > 1$.

For simplicity, we assume $n \triangleq 2p + 1$ odd in the following. The results with $n = 2p$ even are relegated to the Appendix, together with technical proofs.

A.3.1 Circular Seriation with Symmetric, Circulant Matrices

Let us consider the set $\mathcal{C}_R^*$ of matrices in $\mathcal{C}_R$ that are circulant, in order to have a closed form expression of their spectrum. A matrix $A \in \mathbb{R}^{n\times n}$ is Toeplitz if its entries are constant on a given diagonal, $A_{ij} = b_{(i-j)}$ for a vector of values b of size $2n-1$. A symmetric Toeplitz matrix A satisfies $A_{ij} = b_{|i-j|}$, with b of size n. In the case of circulant symmetric matrices, we also have that $b_k = b_{n-k}$, for $1 \leq k \leq n$, thus symmetric circulant matrices are of the form,

$$A = \begin{pmatrix} b_0 & b_1 & b_2 & \cdots & b_2 & b_1 \\ b_1 & b_0 & b_1 & \cdots & b_3 & b_2 \\ b_2 & b_1 & b_0 & \cdots & b_4 & b_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ b_2 & b_3 & b_4 & \cdots & b_0 & b_1 \\ b_1 & b_2 & b_3 & \cdots & b_1 & b_0 \end{pmatrix}. \tag{A.4}$$

Where b is a vector of values of size $p+1$ (recall that $n = 2p+1$). The circular-R assumption (Def A.1.2) imposes that the sequence $(b_0, \ldots, b_{p+1})$ is non-increasing. We thus define the set $\mathcal{C}_R^*$ of circulant matrices of $\mathcal{C}_R$ as follows.

Definition A.3.1. *A matrix $A \in \mathbf{S}^n$ is in $\mathcal{C}_R^*$ iff it verifies $A_{ij} = b_{|i-j|}$ and $b_k = b_{n-k}$ for $1 \leq k \leq n$ with $(b_k)_{k=0,\ldots,\lfloor n/2 \rfloor}$ a non-increasing sequence.*

The spectrum of symmetric circulant matrices is known [Reichel and Trefethen, 1992, Gray et al., 2006, Massey et al., 2007], and for a matrix A of size $n = 2p+1$, it is given by,

$$\begin{array}{lcl} \nu_m & = & b_0 + 2\sum_{k=1}^{p} b_k \cos\left(2\pi km/n\right) \\ y^{m,\cos} & = & \frac{1}{\sqrt{n}}\left(1, \cos\left(2\pi m/n\right), \ldots, \cos\left(2\pi m(n-1)/n\right)\right) \\ y^{m,\sin} & = & \frac{1}{\sqrt{n}}\left(1, \sin\left(2\pi m/n\right), \ldots, \sin\left(2\pi m(n-1)/n\right)\right) \ . \end{array} \tag{A.5}$$

For $m = 1, \ldots, p$, ν_m is an eigenvalue of multiplicity 2 with associated eigenvectors $y^{m,\cos}$,$y^{m,\sin}$. For any m, $(y^{m,\cos}, y^{m,\sin})$ embeds the points on a circle, but for $m > 1$, the circle is walked through m times, hence the ordering of the points on the circle does not follow their latent ordering. The ν_m from equations (A.5) are in general not sorted. It is the Robinson property (monotonicity of (b_k)) that guarantees that $\nu_1 \geq \nu_m$, for $m \geq 1$, and thus that the 2-**LE** embeds the points on a circle *that follows the latent ordering* and allows one to recover it by scanning through the unit circle. This is formalized in Theorem A.3.2, which is the main result of our paper, proved in Appendix A.C. It provides guarantees in the same form as in Theorem A.2.1 with the simple Algorithm 13 that sorts the angles, used in Coifman et al. [2008].

Algorithm 13 Circular Spectral Ordering [Coifman et al., 2008]

Input: Connected similarity matrix $A \in \mathbb{R}^{n \times n}$

1: Compute normalized Laplacian $L_A^{\mathrm{rw}} = \mathbf{I} - (\mathbf{diag}(A\mathbf{1}))^{-1} A$

2: Compute the two first non-trivial eigenvectors of L_A^{rw}, (f_1, f_2)

3: Sort the values of $\theta(i) \triangleq \tan^{-1}(f_2(i)/f_1(i)) + \mathbb{1}[f_1(i) < 0]\pi$

Output: Permutation $\sigma : \theta(\sigma(1)) \leq \ldots \leq \theta(\sigma(n))$

Theorem A.3.2. *Given a permuted observation $\Pi A \Pi^T$ ($\Pi \in \mathcal{P}$) of a matrix $A \in \mathcal{C}_R^*$, the 2-***LE*** maps the items on a circle, equally spaced by angle $2\pi/n$, following the circular ordering in Π. Hence, Algorithm 13 recovers a permutation $\Pi \in \mathcal{P}_n$ such that $\Pi A \Pi^T \in \mathcal{C}_R^*$, i.e., it solves Circular Seriation.*

A.3.2 Perturbation Analysis

The spectrum is a continuous function of the matrix. Let us bound the deformation of the 2-**LE** under a perturbation of the matrix A using the Davis-Kahan theorem [Davis and Kahan, 1970], well introduced in [Von Luxburg, 2007, Theorem 7]. We give more detailed results in Appendix A.D for a subclass of $\mathcal{C}_R^*$ (KMS) defined further.

Proposition A.3.3 (Davis-Kahan)**.** *Let L and $\tilde{L} = L + \delta L$ be the Laplacian matrices of $A \in \mathcal{C}_R^*$ and $A + \delta A \in \mathbf{S}^n$, respectively, and $V, \tilde{V} \in \mathbb{R}^{2 \times n}$ be the associated 2-***LE*** of L and $\tilde{L}$, i.e., the concatenation of the two eigenvectors associated to the two smallest non-zero eigenvalues, written $\lambda_1 \leq \lambda_2$ for L. Then, there exists an orthonormal rotation matrix O such that*

$$\frac{\|V_1 - \tilde{V}_1 O\|_F}{\sqrt{n}} \leq \frac{\|\delta A\|_F}{\min(\lambda_1, \lambda_2 - \lambda_1)} \,. \tag{A.6}$$

A.3.3 Robinson Toeplitz Matrices

Let us investigate how the latent linear ordering of Toeplitz matrices in $\mathcal{L}_R$ translates to the d-**LE**. Remark that from Theorem A.2.1, the 1-**LE** suffices to solve Linear Seriation. Yet, for perturbed observations of $A \in \mathcal{L}_R$, the d-**LE** may be more robust to the perturbation than the 1-**LE**, as the experiments in §A.5 indicate.

Tridiagonal Toeplitz matrices are defined by $b_0 > b_1 > 0 = b_2 = \ldots = b_p$. For $m = 0, \ldots, n-1$, they have eigenvalues ν_m with multiplicity 1 associated to eigenvector $y^{(m)}$ [Trench, 1985],

$$\begin{array}{rcl} \nu_m & = & b_0 + 2b_1 \cos(m\pi/(n+1)) \\ y^{(m)} & = & (\sin(m\pi/(n+1)), \ldots, \sin(mn\pi/(n+1)))\,, \end{array} \tag{A.7}$$

thus matching the spectrum of the Laplace operator on a curve with endpoints from §A.2.3 (up to a shift). This type of matrices can indeed be viewed as a limit case with points uniformly sampled on a line with strong similarity decay, leaving only the two nearest neighbors with non-zero similarity.

Kac-Murdock-Szegö (KMS) matrices are defined, for $\alpha > 0$, $\rho = e^{-\alpha}$, by $A_{ij} = b_{|i-j|} = e^{-\alpha|i-j|} = \rho^{|i-j|}$. For $m = 1, \ldots, \lfloor n/2 \rfloor$, there exists $\theta_m \in ((m-1)\pi/n, m\pi/n)$, such that ν_m

is a double eigenvalue associated to eigenvectors $y^{m,\cos}$,$y^{m,\sin}$,

$$\begin{array}{lcl} \nu_m & = & \frac{1-\rho^2}{1-2\rho\cos\theta_m+\rho^2} \\ y^{m,\cos} & = & \left(\cos\left((n-2r+1)\theta_m/2\right)\right)_{r=1}^{n} \\ y^{m,\sin} & = & \left(\sin\left((n-2r+1)\theta_m/2\right)\right)_{r=1}^{n} \ . \end{array} \tag{A.8}$$

Linearly decreasing Toeplitz matrices defined by $A_{ij}^{lin} = b_{|i-j|} = n - |i-j|$ have spectral properties analog to those of KMS matrices (trigonometric expression, interlacement, low frequency assigned to largest eigenvalue), but with more technical details available in Bünger [2014]. This goes beyond the asymptotic case modeled by tridiagonal matrices.

Banded Robinson Toeplitz matrices typically include similarity matrices from DNA sequencing. Actually, any Robinson Toeplitz matrix becomes banded under a thresholding operation. Also, fast decaying Robinson matrices such as KMS matrices are almost banded. There is a rich literature dedicated to the spectrum of generic banded Toeplitz matrices [BoeÓttcher and Grudsky, 2005, Gray et al., 2006, Böttcher et al., 2017]. However, it mostly provides asymptotic results on the spectra. Notably, some results indicate that the eigenvectors of some banded symmetric Toeplitz matrices become, up to a rotation, close to the sinusoidal, almost equi-spaced eigenvectors observed in equations (A.7) and (A.8) [Böttcher et al., 2010, Ekström et al., 2017].

A.3.4 Spectral Properties of the Laplacian

For circulant matrices A, L_A and A have the same eigenvectors since $L_A = \mathbf{diag}(A\mathbf{1}) - A = c\mathbf{I} - A$, with $c \triangleq \sum_{k=0}^{n-1} b_k$. For general symmetric Toeplitz matrices, this property no longer holds as $c_i = \sum_{j=1}^{n} b_{|i-j|}$ varies with i. Yet, for fast decaying Toeplitz matrices, c_i is almost constant except for i at the edges, namely i close to 1 or to n. Therefore, the eigenvectors of L_A resemble those of A except for the "edgy" entries.

A.4 Recovering Ordering on Filamentary Structure

We have seen that (some) similarity matrices A with a latent ordering lead to a filamentary d-**LE**. The d-**LE** integrates local proximity constraints together into a global consistent embedding. We expect isolated (or, uncorrelated) noise on A to be averaged out by the spectral picture. Therefore, we present Algorithm 14 that redefines the similarity S_{ij} between two items from their proximity within the d-**LE**. Basically, it fits the points by a line *locally*, in the same spirit as LLE, which makes sense when the data lies on a linear manifold (curve) embedded in $\mathbb{R}^K$. Note that Spectral Ordering (Algorithm 12) projects all points on a given line (it only looks at the first coordinates $f_1(i)$) to reorder them. Our method does so in a local neighborhood, allowing for reordering points on a curve with several oscillations. We then run the basic Algorithms 12 (or 13 for Circular Seriation). Hence, the d-**LE** is eventually used to pre-process the similarity matrix.

In Algorithm 14, we compute a d-**LE** in line 1 and then a 1-**LE** (resp., a 2-**LE**) for linear ordering (resp., a circular ordering) in line 9. For reasonable number of neighbors k in the k-NN of line 4 (in practice, $k = 10$), the complexity of computing the d-**LE** dominates Algorithm 14. We shall see in Section A.5 that our method, while being almost as computationally cheap as the base Algorithms 12 and 13 (roughly only a factor 2), yields substantial improvements. In

Algorithm 14 Ordering Recovery on Filamentary Structure in $\mathbb{R}^K$.

Input: A similarity matrix $A \in \mathcal{S}_n$, a neighborhood size $k \geq 2$, a dimension of the Laplacian Embedding d.

1: $\Phi = \left(\boldsymbol{y}_1^T, \ldots, \boldsymbol{y}_n^T\right)^T \in \mathbb{R}^{n \times d} \leftarrow$ d-**LE**(A) ▷ Compute Laplacian Embedding
2: Initialize $S = \mathbf{I}_n$ ▷ New similarity matrix
3: **for** $i = 1, \ldots, n$ **do**
4: $V \leftarrow \{j \,:\, j \in k\text{-NN}(\boldsymbol{y}_i)\} \cup \{i\}$ ▷ find k nearest neighbors of $\boldsymbol{y}_i \in \mathbb{R}^d$
5: $w \leftarrow \text{LinearFit}(V)$ ▷ fit V by a line
6: $D_{uv} \leftarrow |w^T(\boldsymbol{y}_u - \boldsymbol{y}_v)|$, for $u, v \in V$. ▷ Compute distances on the line
7: $S_{uv} \leftarrow S_{uv} + D_{uv}^{-1}$, for $u, v \in V$. ▷ Update similarity
8: **end for**
9: Compute σ^* from the matrix S with Algorithm 12 (resp., Algorithm 13) for a linear (resp., circular) ordering.

Output: A permutation σ^*.

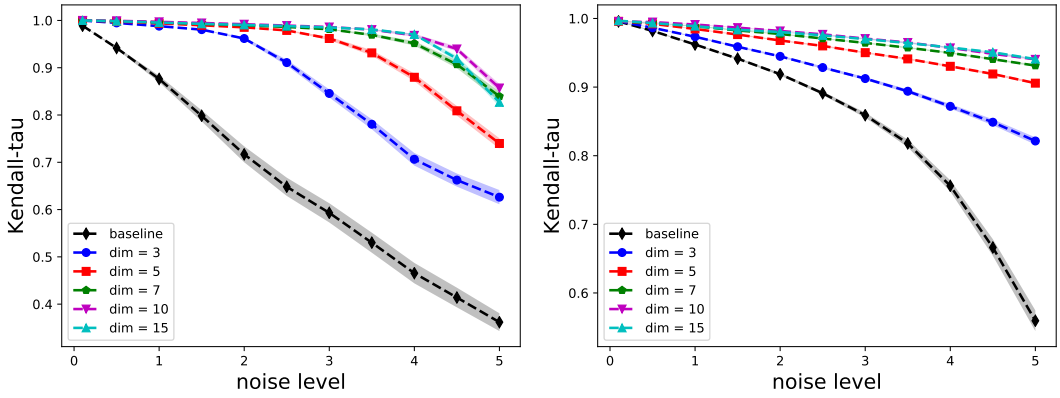
line 7 we can update the similarity S_{uv} by adding any non-increasing function of the distance D_{uv}, *e.g.*, D_{uv}^{-1}, $\exp(-D_{uv})$, or $-D_{uv}$ (the latter case requires to add an offset to S afterwards to ensure it has non-negative entries. It is what we implemented in practice.) In line 9, the matrix S needs to be connected in order to use Algorithm 12, which is not always verified in practice (for low values of k, for instance). In that case, we reorder separately each connected component of S with Algorithm 12, and then merge the partial orderings into a global ordering by using the input matrix A, as detailed in Algorithm 15, Appendix A.A.

A.5 Numerical Results

A.5.1 Synthetic Experiments

We performed synthetic experiments with noisy observations of Toeplitz matrices A, either linear ($\mathcal{L}_R$) or circular ($\mathcal{C}_R^*$). We added a uniform noise on all the entries, with an amplitude parameter a varying between 0 and 5, with maximum value of the noise $a\|A\|_F$. The matrices A used are either banded (sparse), with linearly decreasing entries when moving away from the diagonal, or dense, with exponentially decreasing entries (KMS matrices). We used $n = 500$, several values for the parameters k (number of neighbors) and d (dimension of the d-**LE**), and various scalings of the d-**LE** (parameter α in (α, d)-**LE**), yielding similar results (see sensitivity to the number of neighbors k and to the scaling (α, d)-**LE** in Appendix A.B). In an given experiment, the matrix A is randomly permuted with a ground truth permutation π^*. We report the Kendall-Tau scores between π^* and the solution of Algorithm 14 for different choices of dimension K, for varying noise amplitude a, in Figure A.1, for banded (circular) matrices. For the circular case, the ordering is defined up to a shift. To compute a Kendall-Tau score from two permutations describing a circular ordering, we computed the best Kendall-Tau scores between the first permutation and all shifts from the second, as detailed in Algorithm 16. The analog results for exponentially decaying (KMS) matrices are given in Appendix A.B, Figure A.B.1. For a given combination of parameters, the scores are averaged on 100 experiments and the standard-deviation divided by $\sqrt{n_{\text{exps}}} = 10$ (for ease of reading) is plotted in transparent above and below the curve. The baseline (in blue) corresponds to the basic spectral method of Algorithm 12 for linear and Algorithm 13 for circular seriation. Other lines correspond to given choices of the dimension of the d-**LE**, as written in the legend.

We observe that leveraging the additional dimensions of the d-**LE** unused by the baseline



(a) Linear Banded

(b) Circular Banded

Figure A.1: From left to right, Kendall-Tau scores for Linear and Circular Seriation for noisy observations of banded, Toeplitz, matrices, displayed for several values of the dimension parameter of the d-**LE**(d), for fixed number of neighbors $k = 15$.

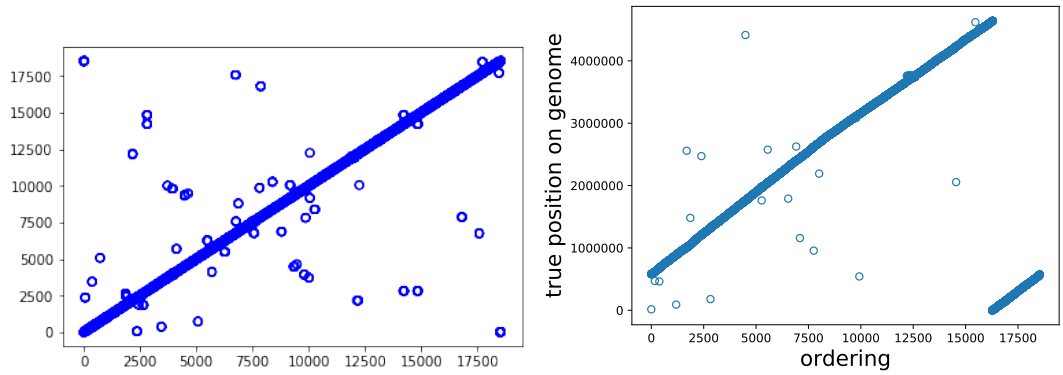
methods Algorithm 12 and 13 substantially improves the robustness of Seriation. For instance, in Figure A.1, the performance of Algorithm 14 is almost optimal for a noise amplitude going from 0 to 4, when it falls by a half for Algorithm 12. We illustrate the effect of the preprocessing of Algorithm 14 in Figures A.B.6 and A.B.7, Appendix A.B.

A.5.2 Genome Assembly Experiments

In *de novo* genome assembly, a whole DNA strand is reconstructed from randomly sampled sub-fragments (called *reads*) whose positions within the genome are unknown. The genome is oversampled so that all parts are covered by multiple reads with high probability. Overlap-Layout-Consensus (OLC) is a major assembly paradigm based on three main steps. First, compute the overlaps between all pairs of read. This provides a similarity matrix A, whose entry (i, j) measures how much reads i and j overlap (and is zero if they do not). Then, determine the layout from the overlap information, that is to say find an ordering and positioning of the reads that is consistent with the overlap constraints. This step, akin to solving a one dimensional jigsaw puzzle, is a key step in the assembly process. Finally, given the tiling of the reads obtained in the layout stage, the consensus step aims at determining the most likely DNA sequence that can be explained by this tiling. It essentially consists in performing multi-sequence alignments.

In the true ordering (corresponding to the sorted reads' positions along the genome), a given read overlaps much with the next one, slightly less with the one after it, and so on, until a point where it has no overlap with the reads that are further away. This makes the read similarity matrix Robinson and roughly band-diagonal (with non-zero values confined to a diagonal band). Finding the layout of the reads therefore fits the Linear Seriation framework (or Circular Seriation for circular genomes). In practice however, there are some repeated sequences (called *repeats*) along the genome that induce false positives in the overlap detection tool [Pop, 2004], resulting in non-zero similarity values outside (and possibly far away) from the diagonal band. The similarity matrix ordered with the ground truth is then the sum of a Robinson band matrix and a sparse "noise" matrix, as in Figure A.2. Because of this sparse

"noise", the basic spectral Algorithm 12 fails to find the layout, as the quadratic loss appearing in 2-SUM is sensitive to outliers. Recanati et al. [2018b] tackle this issue by modifying the loss in 2-SUM to make it more robust. Instead, we show that the simple multi-dimensional extension proposed in Algorithm 14 suffices to capture the ordering of the reads despite the repeats.



(a) similarity matrix

(b) ordering found

Figure A.2: From left to right: Overlap-based similarity matrix from *E. coli* reads, and the ordering found with Algorithm 14 versus the position of the reads within a reference genome obtained by mapping to a reference with minimap2; The genome being circular, the ordering is defined up to a shift, which is why we observe two lines instead of one.

We used our method to perform the layout of a *E. coli* bacterial genome. We used reads sequenced with third-generation sequencing data, and computed the overlaps with dedicated software, as detailed in Appendix A.B. The new similarity matrix S computed from the embedding in Algorithm 14 was disconnected, resulting in several connected component instead of one global ordering. However, the sub-orderings could be unambiguously merged into one in a simple way described in Algorithm 15. The Kendall-Tau score between the ordering found and the one obtained by sorting the position of the reads along the genome (obtained by mapping the reads to a reference with minimap2 [Li, 2018]) is of 99.5%, using Algorithm 16 to account for the circularity of the genome.

A.6 Conclusion

Here, we brought together results that shed light on the filamentary structure of the Laplacian embedding of serial data. It allows for tackling Linear Seriation and Circular Seriation in a unifying framework. Notably, we provide theoretical guarantees for Circular Seriation analog to those existing for Linear Seriation. These do not make assumptions about the underlying generation of the data matrix, and can be verified *a posteriori* by the practitioner. Then, we propose a simple method to leverage the filamentary structure of the embedding. It can be seen as a pre-processing of the similarity matrix. Although the complexity is comparable to the baseline methods, experiments on synthetic and real data indicate that this pre-processing substantially improves robustness to noise.

Appendices

Notation: We will commonly denote σ a permutation of $\{1, \dots, n\}$ and $\mathfrak{S}$ the set of all such permutations. When represented matricially, σ will often be noted Π while cyclic permutation of $\{1, \dots, n\}$ will be noted as τ. A will usually denote the matrix of raw pair-wise similarities. S will denote the similarity matrix resulting from Algorithm 14, and k a neighboring parameter. Finally we use indexed version ν (resp., λ) to denote eigenvalues of a similarity matrix (resp. a graph Laplacian).

A.A Additional Algorithms

A.A.1 Merging Connected Components

The new similarity matrix S computed in Algorithm 14 is not necessarily the adjacency matrix of a connected graph, even when the input matrix A is. For instance, when the number of nearest neighbors k is low and the points in the embedding are non uniformly sampled along a curve, S may have several, disjoint connected components (let us say there are C of them in the following). Still, the baseline Algorithm 12 requires a connected similarity matrix as input. When S is disconnected, we run 12 separately in each of the C components, yielding C sub-orderings instead of a global ordering.

However, since A is connected, we can use the edges of A between the connected components to merge the sub-orderings together. Specifically, given the C ordered subsequences, we build a meta similarity matrix between them as follows. For each pair of ordered subsequences (c_i, c_j), we check whether the elements in one of the two ends of c_i have edges with those in one of the two ends of c_j in the graph defined by A. According to that measure of similarity and to the direction of these meta-edges (*i.e.*, whether it is the beginning or the end of c_i and c_j that are similar), we merge together the two subsequences that are the closest to each other. We repeat this operation with the rest of the subsequences and the sequence formed by the latter merge step, until there is only one final sequence, or until the meta similarity between subsequences is zero everywhere. We formalize this procedure in the greedy Algorithm 15, which is implemented in the package at https://github.com/antrec/mdso.

Given C reordered subsequences (one per connected component of S) $(c_i)_{i=1,\dots,C}$, that form a partition of $\{1, \dots, n\}$, and a window size h that define the length of the ends we consider (h must be smaller than half the smallest subsequence), we denote by c_i^- (resp. c_i^+) the first (resp. the last) h elements of c_i, and $a(c_i^\epsilon, c_j^{\epsilon'}) = \sum_{u \in c_i^\epsilon, v \in c_j^{\epsilon'}} A_{uv}$ is the similarity between the ends c_i^ϵ and $c_j^{\epsilon'}$, for any pair c_i, c_j, $i \neq j \in \{1, \dots, C\}$, and any combination of ends $\epsilon, \epsilon' \in \{+, -\}$. Also, we define the meta-similarity between c_i and c_j by,

$$s(c_i, c_j) \triangleq \max(a(c_i^+, c_j^+), a(c_i^+, c_j^-), a(c_i^-, c_j^+), a(c_i^-, c_j^-)) \ , \tag{A.9}$$

and $(\epsilon_i, \epsilon_j) \in \{+, -\}^2$ the combination of signs where the argmax is realized, *i.e.*, such that $s(c_i, c_j) = a(c_i^{\epsilon_i}, c_j^{\epsilon_j})$. Finally, we will use $\bar{c}_i$ to denote the ordered subsequence c_i read from the end to the beginning, for instance if $c = (1, \dots, n)$, then $\bar{c} = (n, \dots, 1)$.

Algorithm 15 Merging connected components

Input: C ordered subsequences forming a partition $P = (c_1, \dots, c_C)$ of $\{1, \dots, n\}$, an initial similarity matrix A, a neighborhood parameter h.

1: **while** $C > 1$ **do**
2: Compute meta-similarity $\tilde{S}$ such that $\tilde{S}_{ij} = s(c_i, c_j)$, and meta-orientation (ϵ_i, ϵ_j), for all pairs of subsequences with equation A.9.
3: **if** $\tilde{S} = 0$ **then**
4: break
5: **end if**
6: find $(i, j) \in \operatorname{argmax} \tilde{S}$, and (ϵ_i, ϵ_j) the corresponding orientations.
7: **if** $(\epsilon_i, \epsilon_j) = (+, -)$ **then**
8: $c^{\text{new}} \leftarrow (c_i, c_j)$
9: **else if** $(\epsilon_i, \epsilon_j) = (+, +)$ **then**
10: $c^{\text{new}} \leftarrow (c_i, \bar{c}_j)$
11: **else if** $(\epsilon_i, \epsilon_j) = (-, -))$ **then**
12: $c^{\text{new}} \leftarrow (\bar{c}_i, c_j)$
13: **else if** $(\epsilon_i, \epsilon_j) = (-, +))$ **then**
14: $c^{\text{new}} \leftarrow (\bar{c}_i, \bar{c}_j)$
15: **end if**
16: Remove c_i and c_j from P.
17: Add c^{new} to P.
18: $C \leftarrow C - 1$
19: **end while**

Output: Total reordered sequence c^{final}, which is a permutation if $C = 1$ or a set of reordered subsequences if the loop broke at line 5.

A.A.2 Computing Kendall-Tau Score Between Two Permutations Describing a Circular Ordering

Suppose we have data having a circular structure, *i.e.*, we have n items that can be laid on a circle such that the higher the similarity between two elements is, the closer they are on the circle. Then, given an ordering of the points that respects this circular structure (*i.e.*, a solution to Circular Seriation), we can shift this ordering without affecting the circular structure. For instance, in Figure A.A.1, the graph has a $\mathcal{C}_R$ affinity matrix whether we use the indexing printed in black (outside the circle), or a shifted version printed in purple (inside the circle). Therefore, we transpose the Kendall-Tau score between two permutations to the case where we want to compare the two permutations up to a shift with Algorithm 16

Algorithm 16 Comparing two permutation defining a circular ordering

Input: Two permutations vectors of size n, $\sigma = (\sigma(1), \dots, \sigma(n))$ and $\pi = (\pi(1), \dots, \pi(n))$

1: **for** $i = 1$ **to** n **do**
2: $KT(i) \leftarrow \text{Kendall-Tau}(\sigma, (\pi(i), \pi(i+1), \dots, \pi(n), \pi(1), \dots, \pi(i-1)))$
3: **end for**
4: best score $\leftarrow \max_{i=1,\dots,n} KT(i)$

Output: best score

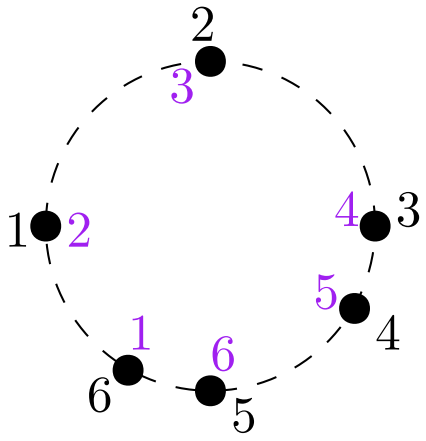
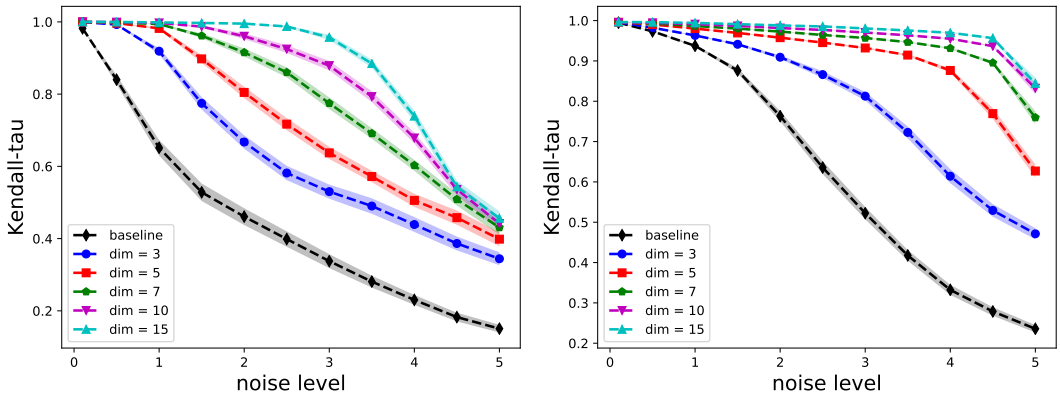


Figure A.A.1: Illustration of the shift-invariance of permutations solution to a Circular Seriation problem.

A.B Additional Numerical Results

Numerical results with KMS matrices In Figure A.B.1 we show the same plots as in Section A.5 but with matrices A such that $A_{ij} = e^{\alpha|i-j|}$, with $\alpha = 0.1$ and $n = 500$.



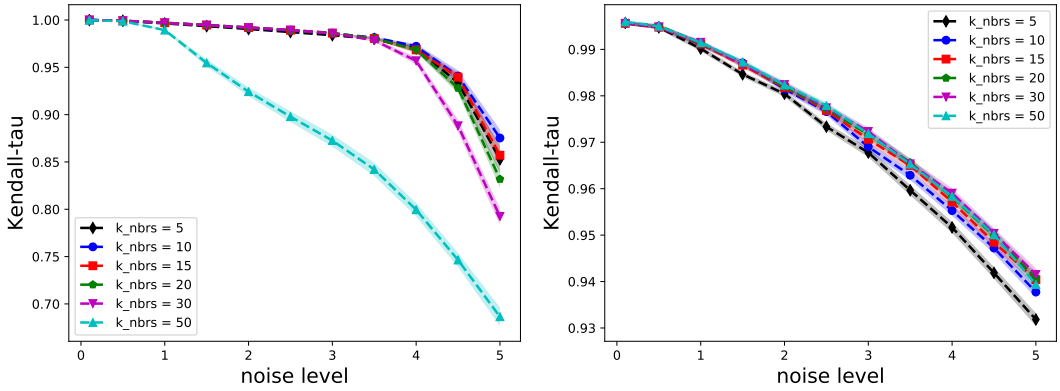
(a) Linear KMS

(b) Circular KMS

Figure A.B.1: From left to right: K-T scores for Linear and Circular Seriation for noisy observations of KMS, Toeplitz, matrices, displayed for several values of the dimension parameter of the d-**LE**.

Sensitivity to parameter k (number of neighbors) Here we show how our method performs when we vary the parameter k (number of neighbors at step 4 of Algorithm 14), for both linearly decrasing, banded matrices, $A_{ij} = \max(c - |i-j|, 0,)$ (as in Section A.5), in Figure A.B.2 and with matrices A such that $A_{ij} = e^{\alpha|i-j|}$, with $\alpha = 0.1$ (Figure A.B.3.

We observe that the method performs roughly equally well with k in a range from 5 to 20, and that the performances drop when k gets too large, around $k = 30$. This can be interpreted as follows. When k is too large, the assumption that the points in the embedding are locally



(a) Linear Banded

(b) Circular Banded

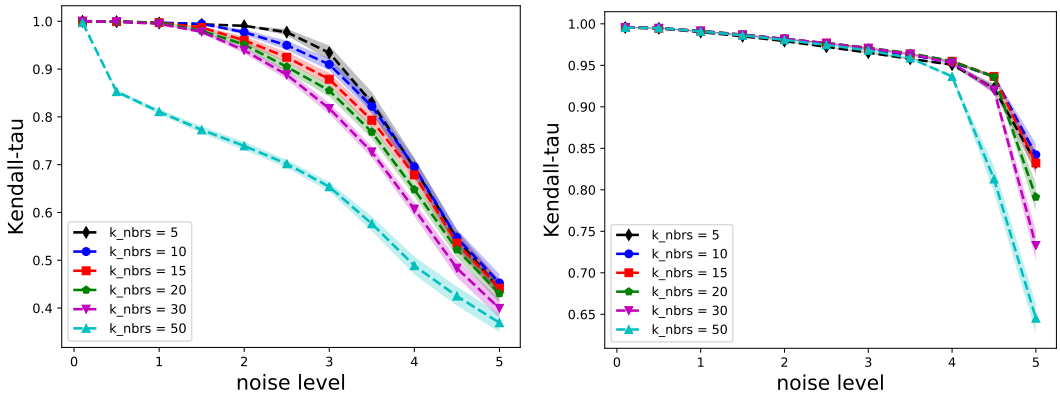
Figure A.B.2: From left to right: K-T scores for Linear and Circular Seriation for noisy observations of banded, Toeplitz, matrices, displayed for several values of the number of nearest neighbors k, with a fixed value of the dimension of the d-**LE**, $d = 10$.

fitted by a line no longer holds. Note also that in practice, for small values of k, *e.g.*, $k = 5$, the new similarity matrix S can be disconnected, and we have to resort to the merging procedure described in Algorithm 15.

Sensitivity to the normalization of the Laplacian We performed experiments to compare the performances of the method with the default Laplacian embedding (d-**LE**) (red curve in Figure A.B.4 and A.B.5) and with two possible normalized embeddings ((α, d)-**LE**) (blue and black curve). We observed that with the default d-**LE**, the performance first increases with d, and then collapses when d gets too large. The CTD scaling (blue) has the same issue, as the first d eigenvalues are roughly of the same magnitude in our settings. The heuristic scaling (α, d)-**LE** with $\alpha_k = 1/\sqrt{k}$ that damps the higher dimensions yields better results when d increases, with a plateau rather than a collapse when d gets large. We interpret these results as follows. With the (d-**LE**), Algorithm 14, line 5 treats equally all dimensions of the embedding. However, the curvature of the embedding tends to increase with the dimension (for $\mathcal{C}_R$ matrix, the period of the cosines increases linearly with the dimension). The filamentary structure is less smooth and hence more sensitive to noise in high dimensions, which is why the results are improved by damping the high dimensions (or using a reasonably small value for d).

Illustration of Algorithm 14 Here we provide some visual illustrations of the method with a circular banded matrix. Given a matrix A (Figure A.B.6), Algorithm 14 computes the d-**LE**. The 2-**LE** is plotted for visualization in Figure A.B.6. Then, it creates a new matrix S (Figure A.B.7) from the local alignment of the points in the d-**LE**. Finally, from the new matrix S, it computes the 2-**LE** (Figure A.B.7), on which it runs the simple method from Algorithm 13.

Figure A.B.6 and A.B.7 give a qualitative illustration of how the method behaves compared to the basic Algorithm 13.



(a) Linear KMS (b) Circular KMS

Figure A.B.3: From left to right: K-T scores for Linear and Circular Seriation for noisy observations of KMS, Toeplitz, matrices, displayed for several values of the number of nearest neighbors k, with a fixed value of the dimension of the d-**LE**, $d = 10$.

A.C Proof of Theorem A.3.2

In this Section, we prove Theorem A.3.2. There are many technical details, notably the distinction between the cases n even and odd. The key idea is to compare the sums involved in the eigenvalues of the circulant matrices $A \in \mathcal{C}_R^*$. It is the sum of the b_k times values of cosines. For λ_1, we roughly have a reordering inequality where the ordering of the b_k matches those of the cosines. For the following eigenvalues, the set of values taken by the cosines is roughly the same, but it does not match the ordering of the b_k. Finally, the eigenvectors of the Laplacian of A are the same than those of A for circulant matrices A, as observed in §A.3.4.

We now introduce a few lemmas that will be useful in the proof.

Notation. In the following we denote $z_k^{(m)} \triangleq \cos(2\pi km/n)$ and $S_p^{(m)} \triangleq \sum_{k=1}^p z_k^{(m)}$. Let's define $\mathcal{Z}_n = \{\cos(2\pi k/n) \mid k \in \mathbb{N}\} \setminus \{-1; 1\}$. Depending on the parity of n, we will write $n = 2p$ or $n = 2p+1$. Hence we always have $p = \lfloor \frac{n}{2} \rfloor$. Also when m and n are not coprime we will note $m = dm'$ as well as $n = dn'$ with n' and m' coprime.

A.C.1 Properties of Sum of Cosinus.

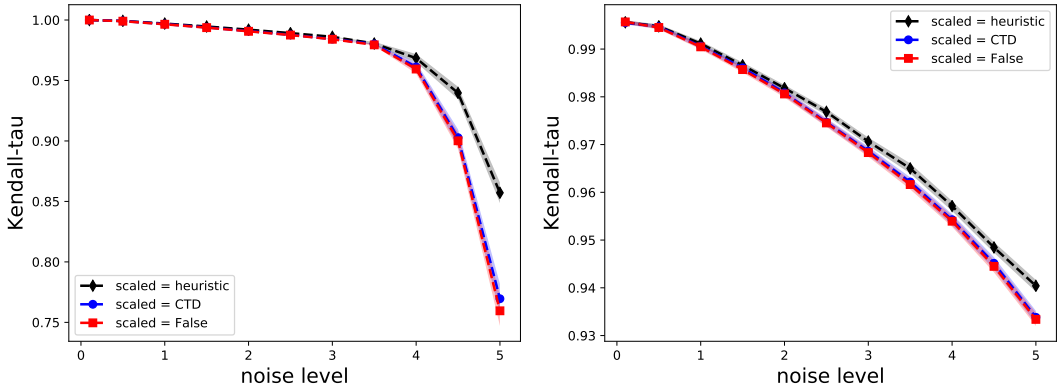
The following lemma gives us how the partial sum sequence $(S_q^{(m)})$ behave for $q = p$ or $q = p-1$ as well as it proves its symmetric behavior in (A.11).

Lemma A.C.1. *For* $z_k^{(m)} = \cos(\frac{2\pi km}{n})$, $n = 2p+1$ *and any* $m = 1, \ldots, p$

$$S_p^{(m)} \triangleq \sum_{k=1}^{p} z_k^{(m)} = -\frac{1}{2} \,. \tag{A.10}$$

Also, for $1 \leq q \leq p/2$,

$$S_{p-q}^{(1)} \geq S_q^{(1)} \,. \tag{A.11}$$



(a) Linear Banded (b) Circular Banded

Figure A.B.4: From left to right: mean of Kendall-Tau for Linear and Circular Seriation for noisy observations of banded, Toeplitz, matrices, displayed for several scalings of the Laplacian embedding, with a fixed number of neighbors $k = 15$ and number of dimensions $d = 10$ in the d-**LE**.

For n and $m \geq 2$ even ($n = 2p$), we have

$$S^{(1)}_{p-1-q} = S^{(1)}_q \quad \textit{for} \quad 1 \leq q \leq (p-1)/2 \tag{A.12}$$

$$S^{(1)}_{p-1} = 0 \quad \textit{and} \quad S^{(m)}_{p-1} = -1 \ . \tag{A.13}$$

Finally for n even and m odd we have

$$S^{(m)}_p = S^{(1)}_p = -1 \ . \tag{A.14}$$

Proof. Let us derive a closed form expression for the cumulative sum $S^{(m)}_q$, for any $m, q \in \{1, \ldots, p\}$

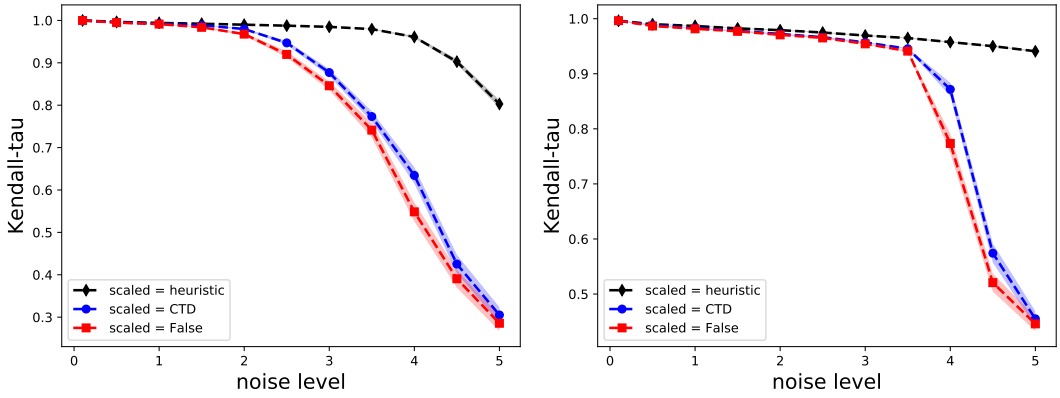
$$\begin{aligned} S^{(m)}_q = \textstyle\sum_{k=1}^q z^{(m)}_k &= \Re\Big(\textstyle\sum_{k=1}^q e^{\frac{2i\pi km}{n}} \Big) \\ &= \Re\Big(e^{2i\pi m/n} \tfrac{1-e^{2i\pi qm/n}}{1-e^{2i\pi m/n}} \Big) \\ &= \cos\left(\pi(q+1)m/n\right) \tfrac{\sin(\pi qm/n)}{\sin(\pi m/n)} \ . \end{aligned} \tag{A.15}$$

Let us prove equation (A.10) with the latter expression for $q = p$. Given that $n = 2p+1 = 2(p+1/2)$, we have,

$$\frac{\pi(p+1)m}{n} = \frac{\pi(p+1/2+1/2)m}{2(p+1/2)} = \frac{\pi m}{2} + \frac{\pi m}{2n},$$
$$\frac{\pi pm}{n} = \frac{\pi(p+1/2-1/2)m}{2(p+1/2)} = \frac{\pi m}{2} - \frac{\pi m}{2n} \ .$$

Now, by trigonometric formulas, we have,

$$\cos\left(\frac{\pi m}{2} + x\right) = \begin{cases} (-1)^{m/2} \cos(x), & \text{if } m \text{ is even} \\ (-1)^{(m+1)/2} \sin(x), & \text{if } m \text{ is odd} \end{cases}$$



(a) Linear Banded (b) Circular Banded

Figure A.B.5: From left to right: mean of Kendall-Tau for Linear and Circular Seriation for noisy observations of banded, Toeplitz, matrices, displayed for several scalings of the Laplacian embedding, with a fixed number of neighbors $k = 15$ and number of dimensions $d = 20$ in the d-**LE**.

$$\sin\left(\frac{\pi m}{2} - x\right) = \begin{cases} (-1)^{(1+m/2)} \sin(x), & \text{if } m \text{ is even} \\ (-1)^{(m-1)/2} \cos(x), & \text{if } m \text{ is odd} \end{cases}$$

It follows that, for any m,

$$\cos\left(\frac{\pi m}{2} + x\right) \sin\left(\frac{\pi m}{2} - x\right) = -\cos(x)\sin(x) = -\frac{1}{2}\sin(2x)$$

Finally, with $x = \pi m/(2n)$, this formula simplifies the numerator appearing in equation (A.15) and yields the result in equation (A.10).

Let us now prove equation (A.11) with a similar derivation. Let $f(q) \triangleq \cos(\pi(q+1)/n)\sin(\pi q/n)$, defined for any real $q \in [1, p/2]$. We wish to prove $f(p-q) \geq f(q)$ for any integer $q \in \{1, \ldots, \lfloor p/2 \rfloor\}$. Using $n = 2(p + 1/2)$, we have,

$$\frac{\pi(p-q+1)}{n} = \frac{\pi(p+1/2-(q-1/2))}{2(p+1/2)} = \frac{\pi}{2} - \frac{\pi(q-1/2)}{n},$$
$$\frac{\pi(p-q)}{n} = \frac{\pi(p+1/2-(q+1/2))}{2(p+1/2)} = \frac{\pi}{2} - \frac{\pi(q+1/2)}{n}.$$

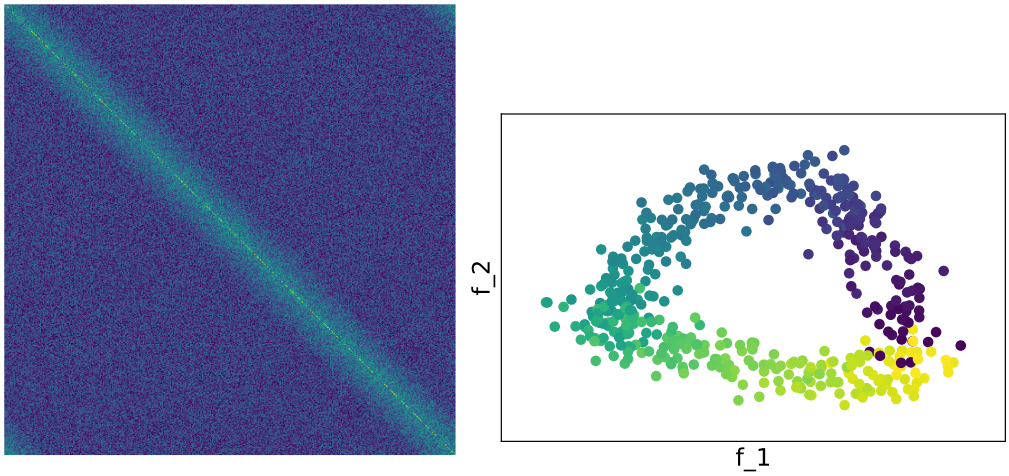
Using $\cos(\pi/2 - x) = \sin(x)$ and $\sin(\pi/2 - x) = \cos(x)$, we thus have,

$$f(p-q) = \cos(\pi(q+1/2)/n)\sin(\pi(q-1/2)/n) = f(q-1/2) \tag{A.16}$$

To conclude, let us observe that $f(q)$ is non-increasing on $[1, p/2]$. Informally, the terms $\{z_k^1\}_{1 \leq k \leq q}$ appearing in the partial sums $S_q^{(1)}$ are all non-negative for $q \leq p/2$. Formally, remark that the derivative of f, $df/dq(q) = (\pi/n)\cos(\pi(2q+1)/n)$ is non-negative for $q \in [1, p/2]$. Hence, for $q \leq p/2$, $f(q-1/2) \geq f(q)$, which ends the proof of equation (A.11).

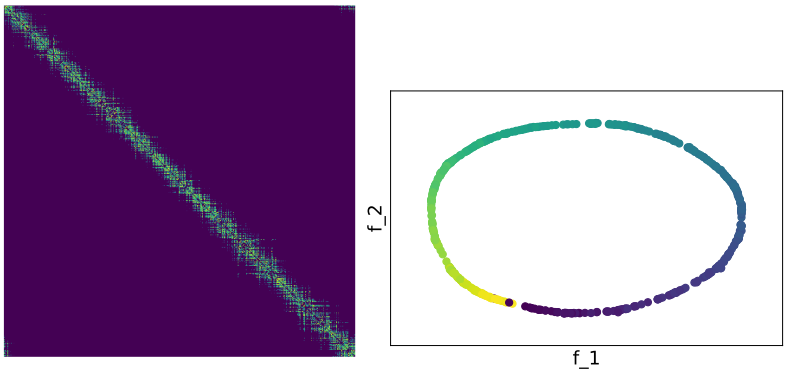
To get the first equality of (A.13), from the exact form in (A.15), we have ($n = 2p$)

$$S_{p-1}^{(1)} = \cos(\pi p/(2p))\frac{\sin(\pi(p-1)/n)}{\sin(\pi/n)} = 0\ .$$



(a) Noisy circular banded matrix A (b) Noisy 2-**LE**

Figure A.B.6: Noisy Circular Banded matrix and associated 2d Laplacian embedding.



(a) Matrix S from Algorithm 14 (b) New 2-**LE**

Figure A.B.7: Matrix S created through Algorithm 14, and associated 2d-Laplacian embedding.

For the second equality in (A.13), we have ($m = 2q$):

$$S^m_{p-1} = \cos(\pi q)\frac{\sin(\pi q - \pi m/n)}{\sin(\pi m/n)} = (-1)^q \frac{-(-1)^q \sin(\pi m/n)}{\sin(\pi m/n)} = -1 \ .$$

Finally to get (A.14), let us write ($n = 2p$ and m odd):

$$\begin{aligned} S^{(m)}_p &= (-1)^{m+1}\frac{\cos(\pi(p+1)m/n)}{\sin(\pi m/n)} = (-1)^{m+1}\frac{\cos(\pi m/2 + \pi m/n)}{\sin(\pi m/n)} \\ &= (-1)^m \sin(\pi m/2) = -1 \ . \end{aligned}$$

■

The following lemma gives an important property of the partial sum of the $z^{(m)}_k$ that is useful when combined with proposition A.C.3.

Lemma A.C.2. *Denote by* $z_k^{(m)} = \cos(2\pi km/n)$. *Consider first* $n = 2p$ *and* m *even. For* $m = 1, \dots, p$ *and* $q = 1, \dots, p-2$

$$S_q^{(1)} = \sum_{k=1}^{q} z_k^{(1)} \geq \sum_{k=1}^{q} z_k^{(m)} = S_q^{(m)} \ . \tag{A.17}$$

Otherwise we have for every $(m, q) \in \{1, \dots, p\}^2$

$$S_q^{(1)} > S_q^{(m)} \ , \tag{A.18}$$

with equality when $q = p$.

Proof. **Case** m **and** n **coprime.** Values of $\left(z_k^{(m)}\right)_{k=1,\dots,p}$ are all distinct. Indeed $z_k^{(m)} = z_{k'}^{(m)}$ implies that n divides $k + k'$ or $k - k'$. It is impossible (the range of $k + k'$ is $[2, 2p]$) unless $k = k'$.

Case m **and** n **not coprime.** $m = dm'$ and $n = dn'$, with $d \geq 3$. In that situation we need to distinguish according to the parity of n.

Case $n = 2p + 1$**.** Let's first remark that $\left(z_k^{(1)}\right)_{k=1,\dots,p}$ takes all values but two (-1 and 1) of the cosinus of multiple of the angle $\frac{2\pi}{n}$, e.g. $\left(z_k^{(1)}\right)_{k=1,\dots,p} \subset \mathcal{Z}_n$. Also $(z_k^{(1)})_{k=1,\dots,p}$ is non-increasing.

Let's prove (A.18) by distinguishing between the various values of q.

- Consider $q = p - (n' - 1), \dots, p$. From (A.10) in lemma (A.C.2), we have $S_p^{(1)} = S_p^{(m)}$. The $\left(z_k^{(1)}\right)_k$ are ordered in non-increasing order and the $\left(z_k^{(m)}\right)_{k=p-n'+1,\dots,p}$ take value in $\mathcal{Z}_n \cup \{1\}$ without repetition (it would requires $k \pm k' \sim 0$ $[n']$). Also the partial sum of $z_k^{(1)}$ starting from the ending point p are lower than any other sequence taking the same or greater value without repetition. Because 1 is largest than any possible value in $\mathcal{Z}_n$, we hence have

$$\sum_{k=q}^{p} z_k^{(1)} \leq \sum_{k=q}^{p} z_k^{(m)} \text{ for any } q = p - (n' - 1), \dots, p \ . \tag{A.19}$$

  Since $S_q^{(m)} = S_p^{(m)} - \sum_{k=q+1}^{p} z_k^{(m)}$, (A.19) implies (A.18) for that particular set of q.

- For $q = 1, \dots, n' - 1$ it is the same type of argument. Indeed the $(z_k^{(1)})_k$ takes the highest values in $\mathcal{Z}_n$ in decreasing order, while $(z_k^{(m)})_k$ takes also its value in $\mathcal{Z}_n$ (because $z_q^{(m)} \neq 1$). This concludes (A.18).

  Note that when $n' \geq \frac{p+1}{2}$, (A.18) is then true for all q. In the sequel, let's then assume that this is not the case, e.g. $n' < \frac{p+1}{2}$.

- For $q = n' - 1, \dots, \lfloor \frac{p}{2} \rfloor$, the $z_q^{(1)}$ are non-negative. Hence $S_q^{(1)}$ is non-decreasing and lower bounded by $S_{n'-1}^{(1)}$. Also because $S_{n'}^{(m)} = 0$ and $S_{n'-1}^{(1)} \geq S_k^{(m)}$ for $k = 1, \dots, n'$, it is true that for all q in the considered set, $S_q^{(m)}$ is upper-bounded by $S_{n'-1}^{(1)}$. All in all it shows (A.18) for these values of q.

- For $q = \lfloor \frac{p}{2} \rfloor + 1, \ldots, p - n'$, we apply (A.11) with $q = n'$ (and indeed $n' \leq \frac{p}{2}$) to get $S^{(1)}_{p-n'} \geq S^{(1)}_{n'}$. Because $S^{(m)}_q$ is upper-bounded by $S^{(1)}_{n'-1}$, it follows that $S^{(1)}_{p-n'} \geq S^{(m)}_q$. Finally since $(S^{(1)}_q)$ is non-increasing for the considered sub-sequence of q, (A.18) is true.

Case $n = 2p$**.** Here $\left(z^{(1)}_k\right)_{k=1,\ldots,p}$ takes unique values in $\mathcal{Z}_n \cup \{-1\}$. We also need to distinguish according to the parity of m.

- $\left(z^{(m)}_k\right)_{k=1,\ldots,n'-1}$ takes also unique value in $\mathcal{Z}_n$. We similarly get (A.18) for $q = 1, \ldots, n' - 1$, and for $q = n'$ because $S^{(m)}_{n'} = 0$.

- Consider m odd, from (A.14), $S^{(m)}_p = S^{(1)}_p = -1$ so that we can do the same reasoning as with n odd to prove (A.18) for $q = p - n' + 1, \ldots, p$ and $q = 1, \ldots, n'$. The remaining follows from the symmetry property (A.12) of the sequence $(S^{(1)}_q)_q$ in Lemma A.C.1.

- m and n even, we have that $S^{(1)}_{p-1} = 0$ and $S^{(m)}_{p-1} = -1$ so that

$$S^{(1)}_{p-1} \geq S^{(m)}_{p-1} + 1 \ .$$

  $S^{(1)}_q \geq S^{(m)}_q$ for $q < p - 1$ follows with same techniques as before.

■

A.C.2 Properties on R-Toeplitz Circular Matrix.

This proposition is a technical method that will be helpful at proving that the eigenvalues of a R-circular Toeplitz matrix are such that $\nu_1 > \nu_m$.

Proposition A.C.3. *Suppose than for any* $k = 1, \ldots, q$ *:*

$$W_k \triangleq \sum_{i=1}^{k} w_i \geq \sum_{i=1}^{k} \tilde{w}_i \triangleq \tilde{W}_k \ ,$$

with (w_i) *and* $(\tilde{w}_i)$ *two sequences of reals. Then, if* $(b_k)_k$ *is non increasing and non negative, we have*

$$\sum_{k=1}^{q} b_k w_k \geq \sum_{k=1}^{q} b_k \tilde{w}_k \ . \tag{A.20}$$

Proof. We have

$$\begin{aligned}
\sum_{k=1}^{q} b_k w_k &= \sum_{k=1}^{q} b_k (W_k - W_{k-1}) \\
&= \underbrace{b_q}_{\geq 0} W_q + \sum_{k=1}^{q-1} \underbrace{(b_k - b_{k+1})}_{\geq 0} W_k \\
&\geq b_q \tilde{W}_q + \sum_{k=1}^{q-1} (b_k - b_{k+1}) \tilde{W}_k = \sum_{k=1}^{q} b_k \tilde{W}_k \ .
\end{aligned}$$

■

As soon as there exists $k_0 \in \{1, \dots, q\}$ such that

$$\sum_{i=1}^{k_0} w_i > \sum_{i=1}^{k_0} \tilde{w}_i \ ,$$

then (A.20) holds strictly.

The following proposition gives the usual derivations of eigenvalues in the R-circular Toeplitz case.

Proposition A.C.4. *Consider A, a circular-R Toeplitz matrix of size n.*

For $n = 2p + 1$

$$\nu_m \triangleq b_0 + 2 \sum_{k=1}^{p} b_k \cos\left(\frac{2\pi k m}{n}\right) . \tag{A.21}$$

For $m = 1, \dots, p$ each ν_m are eigenvalues of A with multiplicity 2 and associated eigenvectors

$$\begin{array}{rcl} y^{m,cos} & = & \frac{1}{\sqrt{n}} \left(1, \cos\left(2\pi m/n\right), \dots, \cos\left(2\pi m(n-1)/n\right)\right) \\ y^{m,sin} & = & \frac{1}{\sqrt{n}} \left(1, \sin\left(2\pi m/n\right), \dots, \sin\left(2\pi m(n-1)/n\right)\right) \ . \end{array} \tag{A.22}$$

For $n = 2p$

$$\nu_m \triangleq b_0 + 2 \textstyle\sum_{k=1}^{p-1} b_k \cos\left(\frac{2\pi k m}{n}\right) + b_p \cos\left(\pi m\right) \ , \tag{A.23}$$

where ν_0 is still singular, with $y^{(0)} = \frac{1}{\sqrt{n}} (1, \dots, 1)$. ν_p also is, with $y^{(p)} = \frac{1}{\sqrt{n}} (+1, -1, \dots, +1, -1)$, and there are $p - 1$ double eigenvalues, for $m = 1, \dots, p - 1$, each associated to the two eigenvectors given in equation (A.22).

Proof. Let us compute the spectrum of a circular-R, symmetric, circulant Toeplitz matrix. From Gray et al. [2006], the eigenvalues are

$$\nu_m = \sum_{k=0}^{n-1} b_k \rho_m^k \ , \tag{A.24}$$

with $\rho_m = \exp(\frac{2i\pi m}{n})$, and the corresponding eigenvectors are,

$$y^{(m)} = \frac{1}{\sqrt{n}} \left(1, e^{-2i\pi m/n}, \dots, e^{-2i\pi m(n-1)/n}\right) \ , \tag{A.25}$$

for $m = 0, \dots, n - 1$.

Case n is odd, with $n = 2p + 1$. Using the symmetry assumption $b_k = b_{n-k}$, and the fact that $\rho_m^{n-k} = \rho_m^n \rho_m^{-k} = \rho_m^{-k}$, it results in real eigenvalues,

$$\begin{array}{rcl} \nu_m & = & b_0 + \sum_{k=1}^{p} b_k \rho_m^k + \sum_{k=p+1}^{n-1} b_k \rho_m^k \\ & = & b_0 + \sum_{k=1}^{p} b_k \rho_m^k + \sum_{k=1}^{p} b_{n-k} \rho_m^{n-k} \\ & = & b_0 + \sum_{k=1}^{p} b_k (\rho_m^k + \rho_m^{-k}) \\ & = & b_0 + 2 \sum_{k=1}^{p} b_k \cos\left(\frac{2\pi k m}{n}\right) . \end{array} \tag{A.26}$$

Observe also that $\nu_{n-m} = \nu_m$, for $m = 1, \ldots, n-1$, resulting in $p+1$ real distinct eigenvalues. ν_0 is singular, whereas for $m = 1, \ldots, p$, ν_m has multiplicity 2, with eigenvectors y^m and y^{n-m}. This leads to the two following real eigenvectors, $y^{m,\cos} = 1/2(y^m + y^{n-m})$ and $y^{m,\sin} = 1/(2i)(y^m - y^{n-m})$

$$\begin{aligned} y^{m,\cos} &= \frac{1}{\sqrt{n}} \left(1, \cos\left(2\pi m/n\right), \ldots, \cos\left(2\pi m(n-1)/n\right)\right) \\ y^{m,\sin} &= \frac{1}{\sqrt{n}} \left(1, \sin\left(2\pi m/n\right), \ldots, \sin\left(2\pi m(n-1)/n\right)\right) \end{aligned} \tag{A.27}$$

Case n is even, with $n = 2p$. A derivation similar to (A.26) yields,

$$\nu_m = b_0 + 2\sum_{k=1}^{p-1} b_k \cos\left(\frac{2\pi km}{n}\right) + b_p \cos\left(\pi m\right) \tag{A.28}$$

ν_0 is still singular, with $y^{(0)} = \frac{1}{\sqrt{n}}(1, \ldots, 1)$, ν_p also is, with $y^{(p)} = \frac{1}{\sqrt{n}}(+1, -1, \ldots, +1, -1)$, and there are $p-1$ double eigenvalues, for $m = 1, \ldots, p-1$, each associated to the two eigenvectors given in equation (A.22).

■

The following proposition is a crucial property of the eigenvalues of a circular Toeplitz matrix. It later ensures that when choosing the second eigenvalues of the laplacian, it will corresponds to the eigenvectors with the lowest period. It is paramount to prove that the latent ordering of the data can be recovered from the curve-like embedding.

Proposition A.C.5. *A circular-R, circulant Toeplitz matrix has eigenvalues $(\nu_m)_{m=0,\ldots,p}$ such that $\nu_1 \geq \nu_m$ for all $m = 2, \ldots, p$ with $n = 2p$ or $n = 2p+1$.*

Proof. Since the shape of the eigenvalues changes with the parity of n, let's again distinguish the cases.

For n odd, $\nu_1 \geq \nu_m$ is equivalent to showing

$$\sum_{k=1}^{p} b_k \cos(2\pi k/n) \geq \sum_{k=1}^{p} b_k \cos(2\pi km/n) \ . \tag{A.29}$$

It is true by combining proposition A.C.3 with lemma A.C.2. The same follows for n even and m odd.

Consider n and m even. We now need to prove that

$$2\sum_{k=1}^{p-1} b_k \cos\left(\frac{2\pi k}{n}\right) - b_p \geq 2\sum_{k=1}^{p-1} b_k \cos\left(\frac{2\pi km}{n}\right) + b_p \ . \tag{A.30}$$

From lemma A.C.2, we have that

$$\sum_{k=1}^{q} z_k^{(1)} \geq \sum_{k=1}^{q} z_k^{(m)} \text{ for } q = 1, \ldots, p-2 \tag{A.31}$$

$$\sum_{k=1}^{p-1} z_k^{(1)} \geq \sum_{k=1}^{p-1} z_k^{(m)} + 1 \ . \tag{A.32}$$

Applying proposition A.C.3 with $w_k = z_k^{(1)}$ and $\tilde{w}_k = z_k^{(m)}$ for $k \leq p-2$ and $\tilde{w}_{p-1} = z_{p-1}^{(m)} + 1$, we get

$$\sum_{k=1}^{p-1} z_k^{(1)} b_k \geq \sum_{k=1}^{p-1} b_k z_k^{(m)} + b_{p-1} \tag{A.33}$$

$$2\sum_{k=1}^{p-1} z_k^{(1)} b_k \geq 2\sum_{k=1}^{p-1} b_k z_k^{(m)} + 2b_p \ . \tag{A.34}$$

The last inequality results from the monotonicity of (b_k) and is equivalent to (A.30). It concludes the proof. ■

Recovering Exactly the Order. Here we provide the proof for Theorem A.3.2.

Theorem A.C.6. *Consider the seriation problem from an observed matrix $\Pi S \Pi^T$, where S is a R-circular Toeplitz matrix. Denote by L the associated graph Laplacian. Then the two dimensional laplacian spectral embedding (*(Lap-Emb) *with d=2) of the items lies ordered and equally spaced on a circle.*

Proof. Denote $A = \Pi S \Pi^T$. The unnormalized Laplacian of A is $L \triangleq \mathrm{diag}(A1) - A$. The eigenspace associated to its second smallest eigenvalue corresponds to that of μ_1 in A. A and S share the same spectrum. Hence the eigenspace of μ_1 in A is composed of the two vectors $\Pi y^{1,sin}$ and $\Pi y^{1,cos}$.

Denote by $(p_i)_{i=1,\ldots,n} \in \mathbb{R}^2$ the 2-**LE**. Each point is parametrized by

$$p_i = (\cos(2\pi\sigma(i)/n), \sin(2\pi\sigma(i)/n)) \ , \tag{A.35}$$

where σ is the permutation represented matricially by Π. ■

A.D Perturbation Analysis

The purpose of the following is to provide guarantees of robustness to the noise with respect to quantities that we will not try to explicit. Some in depths perturbation analysis exists in similar but simpler settings [**?**]. In particular, linking performance of the algorithm while controlling the perturbed embedding is much more challenging than with a one dimensional embedding.

We have performed graph Laplacian re-normalization to make the initial similarity matrix closer to a Toeplitz matrix. Although we cannot hope to obtain exact Toeplitz Matrix. Hence perturbation analysis provide a tool to recollect approximate Toeplitz matrix with guarantees to recover the ordering.

A.D.1 Davis-Kahan

We first characterize how much each point of the new embedding deviate from its corresponding point in the rotated initial set of points. Straightforward application of Davis-Kahan provides a bound on the Frobenius norm that does not grant directly for individual information on the deviation.

Proposition A.D.1 (Davis-Kahan)**.** *Consider L a graph Laplacian of a R-symmetric-circular Toeplitz matrix A. We add a symmetric perturbation matrix H and denote by $\tilde{A} = A + H$ and $\tilde{L}$ the new similarity matrix and graph Laplacian respectively. Denote by $(p_i)_{i=1,\dots,n}$ and $(\tilde{p}_i)_{i=1,\dots,n}$ the 2-***LE*** coming from L and $\tilde{L}$ respectively. Then there exists a cyclic permutation τ of $\{1,\dots,n\}$ such that*

$$\sup_{i=1,\dots,n} ||p_{\tau(i)} - \tilde{p}_i||_2 \leq \frac{2^{3/2}\min(\sqrt{2}||L_H||_2, ||L_H||_F)}{\min(|\lambda_1|, |\lambda_2 - \lambda_1|)} \ , \tag{A.36}$$

where $\lambda_1 < \lambda_2$ are the first non-zeros eigenvalues of L.

Proof. For a matrix $V \in \mathbb{R}^{n\times d}$, denote by

$$||V||_{2,\infty} = \sup_{i=1,\dots,n} ||V_i||_2 \ ,$$

where V_i are the columns of V. Because in $\mathbb{R}^n$ we have $||\cdot||_\infty \leq ||\cdot||_2$, it follows that

$$\begin{aligned} ||V||_{2,\infty} &\leq ||(||V_i||)_{i=1,\dots,n}||_2 = \sqrt{\sum_{i=1}^n ||V_i||_2^2} \\ &\leq ||V||_F \ . \end{aligned}$$

We apply [Yu et al., 2014, Theorem 2] to our perturbed matrix, a simpler version of classical davis-Kahan theorem [Davis and Kahan, 1970].

Let's denote by (λ_1, λ_2) the first non-zeros eigenvalues of L and by V its associated 2-dimensional eigenspace. Similarly denote by $\tilde{V}$ the 2-dimensional eigenspace associated to the first non-zeros eigenvalues of $\tilde{L}$. There exists a rotation matrix $O \in SO_2(\mathbb{R})$ such that

$$||\tilde{V} - VO||_F \leq \frac{2^{3/2}\min(\sqrt{2}||L_H||_2, ||L_H||_F)}{\min(|\lambda_1|, |\lambda_2 - \lambda_1|)} \ . \tag{A.37}$$

In particular we have

$$\begin{aligned} ||\tilde{V} - VO||_{2,\infty} &\leq ||\tilde{V} - VO||_F \\ ||\tilde{V} - VO||_{2,\infty} &\leq \frac{2^{3/2}\min(\sqrt{2}||L_H||_2, ||L_H||_F)}{\min(|\lambda_1|, |\lambda_2 - \lambda_1|)} \end{aligned}$$

Finally because A is a R-symmetric-circular Toeplitz, from Theorem A.3.2, the row of V are n ordered points uniformly sampled on the unit circle. Because applying a rotation is equivalent to translating the angle of these points on the circle. It follows that there exists a cyclic permutation τ such that

$$\sup_{i=1,\dots,n} ||p_i - \tilde{p}_{\tau(i)}||_2 \leq \frac{2^{3/2}\min(\sqrt{2}||L_H||_2, ||L_H||_F)}{\min(|\lambda_1|, |\lambda_2 - \lambda_1|)} \ ,$$

■

A.D.2 Exact Recovery with Noise for Algorithm 13

When all the points remain in a sufficiently small ball around the circle, Algorithm 13 can exactly find the ordering. Let's first start with a geometrical lemma quantifying the radius of the ball around each $(\cos(\theta_k), \sin(\theta_k))$ so that they do not intersect.

Lemma A.D.2. *For $\boldsymbol{x} \in \mathbb{R}^2$ and $\theta_k = 2\pi k/n$ for $k \in \mathbb{N}$ such that*

$$||\boldsymbol{x} - (\cos(\theta_k), \sin(\theta_k))||_2 \leq \sin(\pi/n) \ , \tag{A.38}$$

we have

$$|\theta_x - \theta_k| \leq \pi/n \ ,$$

where $\theta_x = \tan^{-1}(x_1/x_2) + 1[\boldsymbol{x}_1 < 0]\pi$.

Proof. Let $\boldsymbol{x}$ that satisfies (A.38). Let's assume without loss of generality that $\theta_k = 0$ and $\theta_x \geq 0$. Assume also that $x = \boldsymbol{e}_1 + \sin(\pi/n)u_x$ where u is a unitary vector. A x for which θ_x is maximum over these constrained is such that u_x and x are orthonormal.

Parametrize $u_x = (\cos(\gamma), \sin(\gamma))$, because u_x and x are orthonormal, we have $\cos(\gamma) = \sin(-\pi/n)$. Finally since $\theta_x \geq 0$, it follows that $\gamma = \pi/2 + \pi/n$ and hence with elementary geometrical arguments $\theta_x = \pi/n$.

■

Proposition A.D.3 (Exact circular recovery under noise in Algorithm 13)**.** *Consider a matrix $\tilde{A} = \Pi^T A \Pi + H$ with A a $R-$circular Toeplitz (Π is the matrix associated to the permutation σ) and H a symmetric matrix such that*

$$\min(\sqrt{2}||L_H||_2, ||L_H||_F) \leq 2^{-3/2} \sin(\pi/n) \min(|\lambda_1|, |\lambda_2 - \lambda_1|) \ ,$$

where $\lambda_1 < \lambda_2$ are the first non-zeros eigenvalues of the graph Laplacian of $\Pi^T A \Pi$. Denote by $\hat{\sigma}$ the output of Algorithm 13 when having $\tilde{A}$ as input. Then there exists a cyclic permutation τ such that

$$\hat{\sigma} = \sigma^{-1} \circ \tau^{-1} \ . \tag{A.39}$$

Proof. We have

$$\Pi^T \tilde{A} \Pi = A + \Pi^T H \Pi \ .$$

L is the graph Laplacian associated to A and $\tilde{L}$, the one associated to $\tilde{A}$. Denote by $(p_i)_{i=1,\dots,n}$ and $(\tilde{p}_i)_{i=1,\dots,n}$ the 2-**LE** coming from L and $\tilde{L}$ respectively. $(\tilde{p}_{\sigma^{-1}(i)})_{i=1,\dots,n}$ is the 2-**LE** coming from the graph Laplacian of $\Pi^T \tilde{A} \Pi$.

Applying Proposition A.D.1 with $\Pi^T \tilde{A} \Pi$, there exists a cyclic permutation such that

$$\sup_{i=1,\dots,n} ||\tilde{p}_{\sigma^{-1}(i)} - p_{\tau(i)}||_2 < \frac{2^{3/2} \min(\sqrt{2}||L_{H^\pi}||_2, ||L_{H^\pi}||_F)}{\min(|\lambda_1|, |\lambda_2 - \lambda_1|)} \ ,$$

with $H^\pi = \Pi^T H \Pi$, $\lambda_1 < \lambda_2$ the first non zero eigenvalues of A.

Graph Laplacian involve the diagonal matrix D_H. In particular we have that $D_{H^\pi} = \Pi^T D_H \Pi$. For the unnormalized Laplacian, it results in $L_{H^\pi} = \Pi^T L_H \Pi$. We hence have

$$\begin{aligned} \sup_{i=1,\dots,n} ||\tilde{p}_{\sigma(i)} - p_{\tau(i)}||_2 &< \frac{2^{3/2}\min(\sqrt{2}||L_H||_2, ||L_H||_F)}{\min(|\lambda_1|, |\lambda_2 - \lambda_1|)} \\ \sup_{i=1,\dots,n} ||\tilde{p}_i - p_{\tau\circ\sigma^{-1}(i)}||_2 &< \sin(\pi/n) \ . \end{aligned}$$

From Theorem A.3.2, $p_i = \cos(2\pi i/n)$ for all i. It follows that for any i

$$||\tilde{p}_i - \cos(2\pi\tau\circ\sigma(i)/n)||_2 < \sin(\pi/n) \ .$$

Algorithm 13 recovers the ordering by sorting the values of

$$\theta_i = \tan^{-1}(\tilde{p}_i^1/\tilde{p}_i^2) + 1[\tilde{p}_i^1 < 0]\pi \ ,$$

where $\tilde{p}_i = (\tilde{p}_i^1, \tilde{p}_i^2)$. Applying Lemma A.D.2:

$$|\theta_i - 2\pi(\tau\circ\sigma^{-1})(i)/n| < \pi/n \quad \forall i \in \{1,\dots,n\},$$

so that

$$\theta_{\sigma^{-1}\circ\tau^{-1}(1)} \leq \cdots \leq \theta_{\sigma^{-1}\circ\tau^{-1}(n)} \ . \tag{A.40}$$

Finally $\hat{\sigma} = \sigma^{-1}\circ\tau^{-1}$. ■

Appendix B

Interactive painting experiments

Bibliography

Jacob Abernethy, Kevin A Lai, Kfir Y Levy, and Jun-Kun Wang. Faster rates for convex-concave games. *arXiv preprint arXiv:1805.06792*, 2018.

Jacob D Abernethy and Jun-Kun Wang. On frank-wolfe and equilibrium computation. In *Advances in Neural Information Processing Systems*, pages 6584–6593, 2017.

Karim Adiprasito, Imre Bárány, Nabil H. Mustafa, and Tamás Terpai. Theorems of carathéodory, helly, and tverberg without dimension, 2018.

Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4575–4583, 2016.

James A Albano and David W Messinger. Euclidean commute time distance embedding and its application to spectral anomaly detection. In *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XVIII*, volume 8390, page 83902G. International Society for Optics and Photonics, 2012.

Zeyuan Allen-Zhu, Elad Hazan, Wei Hu, and Yuanzhi Li. Linear convergence of a frank-wolfe type algorithm over trace-norm balls. In *Advances in Neural Information Processing Systems*, pages 6191–6200, 2017.

Ingo Althöfer. On sparse approximations to randomized strategies and convex combinations. *Linear Algebra and its Applications*, 199:339–355, 1994.

Jonathan E Atkins and Martin Middendorf. On physical mapping and the consecutive ones property for sparse matrices. *Discrete Applied Mathematics*, 71(1-3):23–40, 1996.

Jonathan E Atkins, Erik G Boman, and Bruce Hendrickson. A spectral algorithm for seriation and the consecutive ones problem. *SIAM Journal on Computing*, 28(1):297–310, 1998.

Hedy Attouch and Jérôme Bolte. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Mathematical Programming*, 116(1-2):5–16, 2009.

Hédy Attouch, Jérôme Bolte, Patrick Redont, and Antoine Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-łojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010.

Hedy Attouch, Jérôme Bolte, and Benar Fux Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods. *Mathematical Programming*, 137(1-2):91–129, 2013.

Hedy Attouch, Giuseppe Buttazzo, and Gârard Michaille. *Variational analysis in Sobolev and BV spaces: applications to PDEs and optimization*, volume 17. Siam, 2014.

AA Auslender and J-P Crouzeix. Global regularity theorems. *Mathematics of Operations Research*, 13 (2):243–253, 1988.

Dominique Azé and Jean-Noël Corvellec. Nonlinear error bounds via a change of function. *Journal of Optimization Theory and Applications*, 172(1):9–32, 2017.

Francis Bach. Duality between subgradient and conditional gradient methods. *SIAM Journal on Optimization*, 25(1):115–129, 2015.

Francis Bach, Simon Lacoste-Julien, and Guillaume Obozinski. On the equivalence between herding and conditional gradient algorithms. *arXiv preprint arXiv:1203.4523*, 2012.

Francis Bach et al. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends® in Machine Learning*, 6(2-3):145–373, 2013.

Maxim V Balashov and Dusan Repovs. Uniformly convex subsets of the hilbert space with modulus of convexity of the second order. *arXiv preprint arXiv:1101.5685*, 2011.

Keith Ball, Eric A Carlen, and Elliott H Lieb. Sharp uniform convexity and smoothness inequalities for trace norms. *Inventiones mathematicae*, 115(1):463–482, 1994.

Rémi Bardenet, Odalric-Ambrym Maillard, et al. Concentration inequalities for sampling without replacement. *Bernoulli*, 21(3):1361–1385, 2015.

Siddharth Barman. Approximating nash equilibria and dense subgraphs via an approximate version of carathéodory's theorem. *arXiv preprint arXiv:1406.2296*, 2014.

Stephen T Barnard, Alex Pothen, and Horst Simon. A spectral algorithm for envelope reduction of sparse matrices. *Numerical linear algebra with applications*, 2(4):317–334, 1995.

Earl R Barnes. A geometrically convergent algorithm for solving optimal control problems. *SIAM Journal on Control*, 10(3):434–443, 1972.

Mohammad Ali Bashiri and Xinhua Zhang. Decomposition-invariant conditional gradient for general polytopes with line search. In *Advances in Neural Information Processing Systems*, pages 2690–2700, 2017.

Amir Beck and Shimrit Shtern. Linearly convergent away-step conditional gradient for non-strongly convex functions. *Mathematical Programming*, 164(1-2):1–27, 2017.

Amir Beck and Marc Teboulle. A conditional gradient method with linear rate of convergence for solving convex linear systems. *Mathematical Methods of Operations Research*, 59(2):235–247, 2004.

Amir Beck and Luba Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM journal on Optimization*, 2013.

David Belanger, Dan Sheldon, and Andrew Mccallum. Marginal inference in mrfs using frank-wolfe. In *In NIPS Workshop on Greedy Optimization, Frank-Wolfe and Friends*. Citeseer, 2013.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

Mikhail Belkin and Partha Niyogi. Semi-supervised learning on riemannian manifolds. *Machine learning*, 56(1-3):209–239, 2004.

Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3): 334–334, 1997.

Andrew An Bian, Baharan Mirzasoleiman, Joachim M Buhmann, and Andreas Krause. Guaranteed non-convex optimization: Submodular maximization over continuous domains. *arXiv preprint arXiv:1606.05615*, 2016.

Edward Bierstone and Pierre D Milman. Semianalytic and subanalytic sets. *Publications Mathématiques de l'IHÉS*, 67:5–42, 1988.

Avrim Blum, Sariel Har-Peled, and Benjamin Raichel. Sparse approximation via generating point sets. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 548–557. Society for Industrial and Applied Mathematics, 2016.

Ralph P Boas Jr. Some uniformly convex spaces. *Bulletin of the American Mathematical Society*, 46 (4):304–311, 1940.

Albrecht BoeÓttcher and Sergei M Grudsky. *Spectral properties of banded Toeplitz matrices*, volume 96. Siam, 2005.

Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Weakly supervised action labeling in videos under ordering constraints. In *European Conference on Computer Vision*, pages 628–643. Springer, 2014.

Piotr Bojanowski, Rémi Lajugie, Edouard Grave, Francis Bach, Ivan Laptev, Jean Ponce, and Cordelia Schmid. Weakly-supervised alignment of video with text. In *Proceedings of the IEEE international conference on computer vision*, pages 4462–4470, 2015.

Jérôme Bolte, Aris Daniilidis, and Adrian Lewis. The łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17 (4):1205–1223, 2007.

Jérôme Bolte, Aris Daniilidis, Olivier Ley, and Laurent Mazet. Characterizations of łojasiewicz inequalities: subgradient flows, talweg, convexity. *Transactions of the American Mathematical Society*, 362 (6):3319–3363, 2010.

Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2):459–494, 2014.

Jérôme Bolte, Trong Phong Nguyen, Juan Peypouquet, and Bruce W Suter. From error bounds to the complexity of first-order descent methods for convex functions. *Mathematical Programming*, 165(2): 471–507, 2017.

Albrecht Böttcher, Sergei M Grudsky, and Egor A Maksimenko. On the structure of the eigenvectors of large hermitian toeplitz band matrices. In *Recent Trends in Toeplitz and Pseudodifferential Operators*, pages 15–36. Springer, 2010.

Albrecht Böttcher, Johan Manuel Bogoya, SM Grudsky, and Egor Anatol'evich Maximenko. Asymptotics of eigenvalues and eigenvectors of toeplitz matrices. *Sbornik: Mathematics*, 208(11):1578, 2017.

Jean Bourgain, Alain Pajor, Stanislaw J Szarek, and Nicole Tomczak-Jaegermann. On the duality problem for entropy numbers of operators. In *Geometric aspects of functional analysis*, pages 50–63. Springer, 1989.

Jean Bourgain, Sjoerd Dirksen, and Jelani Nelson. Toward a unified theory of sparse dimensionality reduction in euclidean space. *Geometric and Functional Analysis*, 25(4):1009–1088, 2015.

Nicholas Boyd, Geoffrey Schiebinger, and Benjamin Recht. The alternating descent conditional gradient method for sparse inverse problems. *SIAM Journal on Optimization*, 27(2):616–639, 2017.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Gábor Braun, Sebastian Pokutta, and Daniel Zink. Lazifying conditional gradient algorithms. *Proceedings of ICML*, 2017a.

Gábor Braun, Sebastian Pokutta, and Daniel Zink. Lazifying conditional gradient algorithms. In *International Conference on Machine Learning*, pages 566–575, 2017b.

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

Kristian Bredies and Hanna Katriina Pikkarainen. Inverse problems in spaces of measures. *ESAIM: Control, Optimisation and Calculus of Variations*, 19(1):190–218, 2013.

Kristian Bredies, Dirk A Lorenz, and Peter Maass. A generalized conditional gradient method and its connection to an iterative shrinkage method. *Computational Optimization and Applications*, 42(2): 173–193, 2009.

F Bünger. Inverses, determinants, eigenvalues, and eigenvectors of real symmetric toeplitz matrices with linearly increasing entries. *Linear Algebra and its Applications*, 459:595–619, 2014.

James Burke and Sien Deng. Weak sharp minima revisited part i: basic theory. *Control and Cybernetics*, 31:439–469, 2002.

James V Burke and Michael C Ferris. Weak sharp minima in mathematical programming. *SIAM Journal on Control and Optimization*, 31(5):1340–1359, 1993.

Vivien Cabannes, Thomas Kerdreux, Louis Thiry, Tina Campana, and Charly Ferrandes. Dialog on a canvas with a machine. *arXiv preprint arXiv:1910.04386*, 2019.

Michael D Canon and Clifton D Cullum. A tight upper bound on the rate of convergence of frank-wolfe algorithm. *SIAM Journal on Control*, 6(4):509–516, 1968.

Alejandro Carderera and Sebastian Pokutta. Second-order conditional gradients. *arXiv preprint arXiv:2002.08907*, 2020.

Bernd Carl. Inequalities of bernstein-jackson-type and the degree of compactness of operators in banach spaces. *Annales de l'Institut Fourier*, 35(3):79–118, 1985.

Jitender Cheema, TH Noel Ellis, and Jo Dicks. Thread mapper studio: a novel, visual web server for the estimation of genetic linkage maps. *Nucleic acids research*, 38(suppl_2):W188–W193, 2010.

Zaiyi Chen, Yi Xu, Enhong Chen, and Tianbao Yang. Sadagrad: Strongly adaptive stochastic gradient methods. In *International Conference on Machine Learning*, pages 912–920, 2018.

Yiu-ming Cheung and Jian Lou. Efficient generalized conditional gradient with gradient sliding for composite optimization. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

Fan Chung and S-T Yau. Discrete green's functions. *Journal of Combinatorial Theory, Series A*, 91 (1-2):191–214, 2000.

James A Clarkson. Uniformly convex spaces. *Transactions of the American Mathematical Society*, 40 (3):396–414, 1936.

Kenneth L Clarkson. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Transactions on Algorithms (TALG)*, 6(4):63, 2010a.

Kenneth L Clarkson, Elad Hazan, and David P Woodruff. Sublinear optimization for machine learning. *Journal of the ACM (JACM)*, 59(5):1–49, 2012.

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

Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

Ronald R Coifman, Yoel Shkolnisky, Fred J Sigworth, and Amit Singer. Graph laplacian tomography from unknown random projections. *IEEE Transactions on Image Processing*, 17(10):1891–1899, 2008.

Cyrille W Combettes and Sebastian Pokutta. Revisiting the approximate carath\'eodory problem via the frank-wolfe algorithm. *arXiv preprint arXiv:1911.04415*, 2019.

Cyrille W. Combettes and Sebastian Pokutta. Boosting frank-wolfe by chasing gradients, 2020.

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865, 2016.

Dong Dai, Philippe Rigollet, Lucy Xia, Tong Zhang, et al. Aggregation of affine estimators. *Electronic Journal of Statistics*, 8(1):302–327, 2014.

S Damla Ahipasaoglu, Peng Sun, and Michael J Todd. Linear convergence of a modified frank–wolfe algorithm for computing minimum-volume enclosing ellipsoids. *Optimisation Methods and Software*, 23(1):5–19, 2008.

Chandler Davis and William Morton Kahan. The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

Ofer Dekel, Nika Haghtalab, Patrick Jaillet, et al. Online learning with a hint. In *Advances in Neural Information Processing Systems*, pages 5299–5308, 2017.

V. F. Demyanov and A. M. Rubinov. Approximate methods in optimization problems. *Modern Analytic and Computational Methods in Science and Mathematics*, 1970.

Quentin Denoyelle, Vincent Duval, Gabriel Peyré, and Emmanuel Soubies. The sliding frank–wolfe algorithm and its application to super-resolution microscopy. *Inverse Problems*, 36(1):014001, 2019.

Persi Diaconis, Sharad Goel, and Susan Holmes. Horseshoes in multidimensional scaling and local kernel methods. *The Annals of Applied Statistics*, pages 777–807, 2008.

Jacques Dixmier. Formes linéaires sur un anneau d'opérateurs. *Bulletin de la Société Mathématique de France*, 81:9–39, 1953.

Michael J Donahue, Christian Darken, Leonid Gurvits, and Eduardo Sontag. Rates of convex approximation in non-hilbert spaces. *Constructive Approximation*, 13(2):187–220, 1997a.

Michael J Donahue, Christian Darken, Leonid Gurvits, and Eduardo Sontag. Rates of convex approximation in non-hilbert spaces. *Constructive Approximation*, 13(2):187–220, 1997b.

Miroslav Dudik, Zaid Harchaoui, and Jérôme Malick. Lifted coordinate descent for learning with trace-norm regularization. In *Artificial Intelligence and Statistics*, pages 327–336, 2012.

JC Dunn. A simple averaging process for approximating the solutions of certain optimal control problems. *Journal of Mathematical Analysis and Applications*, 48(3):875–894, 1974.

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

Joseph C Dunn. Convergence rates for conditional gradient sequences generated by implicit step length rules. *SIAM Journal on Control and Optimization*, 18(5):473–487, 1980.

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

Alexandre d'Aspremont and Igor Colin. An approximate shapley-folkman theorem. *arXiv preprint arXiv:1712.08559*, 2017.

Jack Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Optimization—Eureka, You Shrink!*, pages 11–26. Springer, 2003.

Sven-Erik Ekström, Carlo Garoni, and Stefano Serra-Capizzano. Are the eigenvalues of banded symmetric toeplitz matrices known in almost closed form? *Experimental Mathematics*, pages 1–10, 2017.

Olivier Fercoq and Zheng Qu. Restarting accelerated gradient methods with a rough strong convexity estimate. *arXiv preprint arXiv:1609.07358*, 2016.

Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. Mind the duality gap: safer rules for the lasso. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

Fajwel Fogel, Rodolphe Jenatton, Francis Bach, and Alexandre d'Aspremont. Convex relaxations for permutation problems. In *Advances in Neural Information Processing Systems*, pages 1016–1024, 2013.

Rina Foygel, Nathan Srebro, and Russ R Salakhutdinov. Matrix reconstruction with the local max norm. In *Advances in Neural Information Processing Systems*, pages 935–943, 2012.

Emanuele Frandi, Ricardo Ñanculef, and Johan Suykens. Complexity issues and randomization strategies in frank-wolfe algorithms for machine learning. *arXiv preprint arXiv:1410.4062*, 2014.

Emanuele Frandi, Ricardo Ñanculef, Stefano Lodi, Claudio Sartori, and Johan A. K. Suykens. Fast and scalable lasso via stochastic Frank–Wolfe methods with a convergence guarantee. *Machine Learning*, 2016.

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

Pierre Frankel, Guillaume Garrigos, and Juan Peypouquet. Splitting methods with variable metric for kurdyka–łojasiewicz functions and general convergence rates. *Journal of Optimization Theory and Applications*, 165(3):874–900, 2015.

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

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

Michael Friendly. Corrgrams: Exploratory displays for correlation matrices. *The American Statistician*, 56(4):316–324, 2002.

Futoshi Futami, Zhenghang Cui, Issei Sato, and Masashi Sugiyama. Bayesian posterior approximation via greedy particle optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3606–3613, 2019.

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

Dan Garber and Elad Hazan. Playing non-linear games with linear oracles. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 420–428. IEEE, 2013b.

Dan Garber and Elad Hazan. Faster rates for the frank-wolfe method over strongly-convex sets. In *Proceedings of the 32th International Conference on Machine Learning (ICML-15)*, 2015.

Dan Garber and Ofer Meshi. Linear-memory and decomposition-invariant linearly convergent conditional gradient algorithm for structured polytopes. *arXiv preprint, arXiv:1605.06492v1*, May 2016.

Dan Garber, Shoham Sabach, and Atara Kaplan. Fast generalized conditional gradient method with applications to matrix recovery problems. *arXiv preprint arXiv:1802.05581*, 2018.

Gemma C Garriga, Esa Junttila, and Heikki Mannila. Banded structure in binary matrices. *Knowledge and information systems*, 28(1):197–226, 2011.

Joachim Giesen, Martin Jaggi, and Sören Laue. Optimizing over the growing spectrahedron. In *European Symposium on Algorithms*, pages 503–514. Springer, 2012.

Elmer G Gilbert. An iterative procedure for computing the minimum of a quadratic form on a convex set. *SIAM Journal on Control*, 4(1):61–80, 1966.

Pontus Giselsson and Stephen Boyd. Monotonicity and restart in fast gradient methods. In *53rd IEEE Conference on Decision and Control*, pages 5058–5063. IEEE, 2014a.

Pontus Giselsson and Stephen Boyd. Monotonicity and restart in fast gradient methods. In *53rd IEEE Conference on Decision and Control*, pages 5058–5063. IEEE, 2014b.

Donald Goldfarb, Garud Iyengar, and Chaoxu Zhou. Semi-stochastic frank-wolfe algorithms with away-steps for block-coordinate structure problems, 2016.

Donald Goldfarb, Garud Iyengar, and Chaoxu Zhou. Linear convergence of stochastic frank wolfe variants. *arXiv preprint arXiv:1703.07269*, 2017.

Vladimir V Goncharov and Grigorii E Ivanov. Strong and weak convexity of closed sets in a hilbert space. In *Operations research, engineering, and cyber security*, pages 259–297. Springer, 2017.

Robert M Gray et al. Toeplitz and circulant matrices: A review. *Foundations and Trends® in Communications and Information Theory*, 2(3):155–239, 2006.

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

David H Gutman and Javier F Pena. The condition of a function relative to a polytope. *arXiv preprint arXiv:1802.00271*, 2018.

Olof Hanner. On the uniform convexity of lp and lp. *Ark. Mat.*, 3(3):239–244, 02 1956.

Zaid Harchaoui, Matthijs Douze, Mattis Paulin, Miroslav Dudik, and Jérôme Malick. Large-scale image classification with trace-norm regularization. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3386–3393. IEEE, 2012.

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

Hamed Hassani, Mahdi Soltanolkotabi, and Amin Karbasi. Gradient methods for submodular maximization. In *Advances in Neural Information Processing Systems*, pages 5841–5851, 2017.

Hamed Hassani, Amin Karbasi, Aryan Mokhtari, and Zebang Shen. Stochastic conditional gradient++, 2019.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Unsupervised learning. In *The elements of statistical learning*, pages 485–585. Springer, 2009.

Elad Hazan and Satyen Kale. Projection-free online learning. *arXiv preprint arXiv:1206.4657*, 2012.

Elad Hazan and Haipeng Luo. Variance-reduced and projection-free stochastic optimization. In *International Conference on Machine Learning*, pages 1263–1271, 2016.

Elad Hazan and Edgar Minasyan. Faster projection-free online learning. *arXiv:2001.11568*, 2020.

Donald W Hearn, S Lawphongpanich, and Jose A Ventura. Restricted simplicial decomposition: Computation and extensions. In *Computation Mathematical Programming*, pages 99–118. Springer, 1987.

Brandon W Higgs, Jennifer Weller, and Jeffrey L Solka. Spectral embedding finds meaningful (relevant) structure in image and microarray data. *Bmc Bioinformatics*, 7(1):74, 2006.

Alan J Hoffman. On approximate solutions of systems of linear inequalities. *Journal of Research of the National Bureau of Standards*, 49(4), 1952.

Charles A Holloway. An extension of the frank and wolfe method of feasible directions. *Mathematical Programming*, 6(1):14–27, 1974.

Ruitong Huang, Tor Lattimore, András György, and Csaba Szepesvari. Following the leader and fast rates in linear prediction: Curved constraint sets and other regularities. In *Advances in Neural Information Processing Systems 29*, pages 4970–4978. 2016a.

Ruitong Huang, Tor Lattimore, András György, and Csaba Szepesvári. Following the leader and fast rates in linear prediction: Curved constraint sets and other regularities. In *Advances in Neural Information Processing Systems*, pages 4970–4978, 2016b.

Ruitong Huang, Tor Lattimore, András György, and Csaba Szepesvári. Following the leader and fast rates in online linear prediction: Curved constraint sets and other regularities. *The Journal of Machine Learning Research*, 18(1):5325–5355, 2017.

Grigory Ivanov. Approximate carathéodory's theorem in uniformly smooth banach spaces. *Discrete & Computational Geometry*, pages 1–8, 2019.

Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. Group lasso with overlap and graph lasso. In *Proceedings of the 26th annual international conference on machine learning*, pages 433–440, 2009.

M. Jaggi. Convex optimization without projection steps. *Arxiv preprint arXiv:1108.1170*, 2011.

Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th international conference on machine learning*, pages 427–435, 2013.

Martin Jaggi, Marek Sulovsk, et al. A simple algorithm for nuclear norm regularized problems. In *Proceedings of the 27th International Conference on Machine Learning*, 2010.

Bradley R Jones, Ashok Rajaraman, Eric Tannier, and Cedric Chauve. Anges: reconstructing ancestral

genomes maps. *Bioinformatics*, 28(18):2388–2390, 2012.

Armand Joulin, Kevin Tang, and Li Fei-Fei. Efficient image and video co-localization with frank-wolfe algorithm. In *European Conference on Computer Vision*, pages 253–268. Springer, 2014.

Michel Journée, Yurii Nesterov, Peter Richtárik, and Rodolphe Sepulchre. Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research*, 11(Feb):517–553, 2010.

Anatoli Juditsky and Arkadii S Nemirovski. Large deviations of vector-valued martingales in 2-smooth normed spaces. *arXiv preprint arXiv:0809.0813*, 2008.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016a.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016b.

Henry J Kelley. Method of gradients. In *Mathematics in Science and Engineering*, volume 5, pages 205–254. Elsevier, 1962.

Thomas Kerdreux, Igor Colin, and Alexandre d'Aspremont. An approximate shapley-folkman theorem. *arXiv preprint arXiv:1712.08559v3*, 2017.

Thomas Kerdreux, Alexandre d'Aspremont, and Sebastian Pokutta. Restarting frank-wolfe: Faster rates under hölderian error bounds. *arXiv preprint arXiv:1810.02429v3*, 2018a.

Thomas Kerdreux, Fabian Pedregosa, and Alexandre d'Aspremont. Frank-wolfe with subsampling oracle. In *International Conference on Machine Learning*, pages 2591–2600, 2018b.

Thomas Kerdreux, Alexandre d'Aspremont, and Sebastian Pokutta. Restarting frank-wolfe. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1275–1283, 2019.

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

Thomas Kerdreux, Louis Thiry, and Erwan Kerdreux. Interactive neural style transfer with artists. *arXiv preprint arXiv:2003.06659*, 2020b.

Shimon Kogan, Dimitry Levin, Bryan R Routledge, Jacob S Sagi, and Noah A Smith. Predicting risk from financial reports with regression. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2009.

Joseph B Kruskal and Myron Wish. *Multidimensional scaling*, volume 11. Sage, 1978.

Piyush Kumar and E Alper Yildirim. Minimum-volume enclosing ellipsoids and core sets. *Journal of Optimization Theory and Applications*, 126(1):1–21, 2005.

Piyush Kumar and E Alper Yıldırım. A linearly convergent linear-time first-order algorithm for support vector classification with a core set result. *INFORMS Journal on Computing*, 23(3):377–391, 2011.

Virendra Kumar. A control averaging technique for solving a class of singular optimal control problems. *International Journal of Control*, 23(3):361–380, 1976.

P Kuntz, F Velin, and H Briand. Iterative geometric representations for multi-way partitioning, 2001.

Krzysztof Kurdyka. On gradients of functions definable in o-minimal structures. In *Annales de l'institut Fourier*, volume 48, pages 769–783, 1998.

Simon Lacoste-Julien. Convergence rate of frank-wolfe for non-convex objectives. *arXiv preprint*

arXiv:1607.00345, 2016.

Simon Lacoste-Julien and Martin Jaggi. An affine invariant linear convergence analysis for frank-wolfe algorithms. *arXiv preprint arXiv:1312.7864*, 2013.

Simon Lacoste-Julien and Martin Jaggi. On the global linear convergence of Frank–Wolfe optimization variants. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28, pages 496–504. Curran Associates, Inc., 2015a.

Simon Lacoste-Julien and Martin Jaggi. On the global linear convergence of frank-wolfe optimization variants. In *Advances in Neural Information Processing Systems*, pages 496–504, 2015b.

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 International Conference on Machine Learning*, 2013.

Simon Lacoste-Julien, Fredrik Lindsten, and Francis Bach. Sequential kernel herding: Frank-wolfe optimization for particle filtering. *arXiv preprint arXiv:1501.02056*, 2015.

Stéphane S Lafon. *Diffusion maps and geometric harmonics*. PhD thesis, Yale University PhD dissertation, 2004.

Jean Lafond, Hoi-To Wai, and Eric Moulines. On the online frank-wolfe algorithms for convex and non-convex optimizations. *arXiv preprint arXiv:1510.01171*, 2015.

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

Guanghui Lan and Yi Zhou. Conditional gradient sliding for convex optimization. *Optimization-Online preprint (4605)*, 2014.

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

Guanghui Lan, Sebastian Pokutta, Yi Zhou, and Daniel Zink. Conditional accelerated lazy stochastic gradient descent. *arXiv preprint arXiv:1703.05840*, 2017.

Gilbert Laporte. The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(2):231–247, 1992.

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

Kfir Levy and Andreas Krause. Projection free online learning over smooth sets. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1458–1466, 2019.

Guoyin Li. Global error bounds for piecewise convex polynomials. *Mathematical programming*, 137 (1-2):37–64, 2013.

Guoyin Li and Ting Kei Pong. Calculus of the exponent of kurdyka–łojasiewicz inequality and its applications to linear convergence of first-order methods. *Foundations of computational mathematics*, 18(5):1199–1232, 2018.

Heng Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 1:7, 2018.

Joram Lindenstrauss and Lior Tzafriri. *Classical Banach spaces II: function spaces*, volume 97. Springer Science & Business Media, 2013.

Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):208–220, 2012.

Jie Liu, Dejun Lin, Gurkan Yardimci, and William Noble. Unsupervised embedding of single-cell hi-c data. *bioRxiv*, page 257048, 2018.

Mingrui Liu and Tianbao Yang. Adaptive accelerated gradient converging method under h\"{o} lderian

error bound condition. In *Advances in Neural Information Processing Systems*, pages 3104–3114, 2017.

Zehua Liu, Huazhe Lou, Kaikun Xie, Hao Wang, Ning Chen, Oscar M Aparicio, Michael Q Zhang, Rui Jiang, and Ting Chen. Reconstructing cell cycle pseudo time-series via single-cell transcriptome data. *Nature communications*, 8(1):22, 2017.

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*, 2017a.

Francesco Locatello, Rajiv Khanna, Michael Tschannen, and Martin Jaggi. A unified optimization view on generalized matching pursuit and frank-wolfe. *arXiv preprint arXiv:1702.06457*, 2017b.

Francesco Locatello, Michael Tschannen, Gunnar Rätsch, and Martin Jaggi. Greedy algorithms for cone constrained optimization with convergence guarantees. In *Advances in Neural Information Processing Systems*, pages 773–784, 2017c.

Stanis Lojasiewicz. Ensembles semi-analytiques. *Institut des Hautes Études Scientifiques*, 1965.

Stanislas Łojasiewicz. Division d'une distribution par une fonction analytique de variables réelles. In *Comptes rendus de l'Académie des Sciences de Paris*, pages 683–686, 1958.

Stanislas Łojasiewicz. Une propriété topologique des sous-ensembles analytiques réels. In *Les équations aux dérivées partielles*, pages 87–89, 1963.

Stanislas Łojasiewicz. Sur la géométrie semi-et sous-analytique. *Ann. Inst. Fourier*, 43(5):1575–1595, 1993.

L Lovasz. Matching theory (north-holland mathematics studies), 1986.

Haihao Lu and Robert M Freund. Generalized stochastic frank–wolfe algorithm with stochastic "substitute" gradient for structured convex optimization. *Mathematical Programming*, pages 1–33, 2020.

Giulia Luise, Saverio Salzo, Massimiliano Pontil, and Carlo Ciliberto. Sinkhorn barycenters with free support via frank-wolfe algorithm. In *Advances in Neural Information Processing Systems*, pages 9318–9329, 2019.

Zhi-Quan Luo and Jos F Sturm. Error bounds for quadratic systems. In *High performance optimization*, pages 383–404. Springer, 2000.

James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

Olvi L Mangasarian. A condition number for differentiable convex inequalities. *Mathematics of Operations Research*, 10(2):175–179, 1985.

Adam Massey, Steven J Miller, and John Sinsheimer. Distribution of eigenvalues of real symmetric palindromic toeplitz matrices and circulant matrices. *Journal of Theoretical Probability*, 20(3):637–662, 2007.

João Meidanis, Oscar Porto, and Guilherme P Telles. On the consecutive ones property. *Discrete Applied Mathematics*, 88(1):325–354, 1998.

Antoine Miech, Jean-Baptiste Alayrac, Piotr Bojanowski, Ivan Laptev, and Josef Sivic. Learning from video and text via large-scale discriminative clustering. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5267–5276. IEEE, 2017.

Antoine Miech, Ivan Laptev, and Josef Sivic. Learning a text-video embedding from incomplete and heterogeneous data. *arXiv preprint arXiv:1804.02516*, 2018.

Athanasios Migdalas. A regularization of the frank—wolfe method and unification of certain nonlinear programming methods. *Mathematical Programming*, 65(1-3):331–345, 1994.

Vahab Mirrokni, Renato Paes Leme, Adrian Vladu, and Sam Chiu-wai Wong. Tight bounds for approximate carath\'eodory and beyond. *arXiv preprint arXiv:1512.08602*, 2015.

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

Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Conditional gradient method for stochastic submodular maximization: Closing the gap. *arXiv preprint arXiv:1711.01660*, 2017.

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

Marco Molinaro. Curvature of feasible sets in offline and online optimization. *arXiv:2002.03213*, 2020.

Ricardo Ñanculef, Emanuele Frandi, Claudio Sartori, and Héctor Allende. A novel frank–wolfe algorithm. analysis and applications to large-scale svm training. *Information Sciences*, 285:66–99, 2014.

Geoffrey Négiar, Gideon Dresdner, Alicia Tsai, Laurent El Ghaoui, Francesco Locatello, and Fabian Pedregosa. Stochastic frank-wolfe for constrained finite-sum minimization. *arXiv preprint arXiv:2002.11860*, 2020.

A. Nemirovskii and Y. Nesterov. Optimal methods of smooth convex minimization. *USSR Computational Mathematics and Mathematical Physics*, pages 21–30, 1985a.

AS Nemirovskii and Yu E Nesterov. Optimal methods of smooth convex minimization. *USSR Computational Mathematics and Mathematical Physics*, 25(2):21–30, 1985b.

AS Nemirovskii and Yu E Nesterov. Optimal methods of smooth convex minimization. *USSR Computational Mathematics and Mathematical Physics*, 25(2):21–30, 1985c.

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

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

Hao Nguyen and Guergana Petrova. Greedy strategies for convex optimization. *Calcolo*, 54(1):207–224, 2017.

Trong Phong Nguyen. A stroll in the jungle of error bounds. *arXiv preprint arXiv:1704.06938*, 2017.

Vlad Niculae, André FT Martins, Mathieu Blondel, and Claire Cardie. Sparsemap: Differentiable sparse structured inference. *arXiv preprint arXiv:1802.04223*, 2018.

Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

Albert B Novikoff. On convergence proofs for perceptrons. Technical report, Stanford research inst menlo park ca, 1963.

Guillaume Obozinski, Laurent Jacob, and Jean-Philippe Vert. Group lasso with overlaps: the latent group lasso approach. *arXiv preprint arXiv:1110.0413*, 2011.

Anton Osokin, Jean-Baptiste Alayrac, Isabella Lukasewitz, Puneet K Dokania, and Simon Lacoste-Julien. Minding the gaps for block frank-wolfe optimization of structured svms. *ICML 2016 International Conference on Machine Learning / arXiv preprint arXiv:1605.09346*, 2016.

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

François-Pierre Paty and Marco Cuturi. Subspace robust wasserstein distances. *arXiv preprint arXiv:1901.08949*, 2019.

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

Javier Pena and Daniel Rodriguez. Polytope conditioning and linear convergence of the frank–wolfe algorithm. *Mathematics of Operations Research*, 2018.

Julia Peyre, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Weakly-supervised learning of visual relations. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5179–5188, 2017.

Iosif Pinelis. Optimum bounds for the distributions of martingales in banach spaces. *The Annals of Probability*, pages 1679–1706, 1994.

Wei Ping, Qiang Liu, and Alexander T Ihler. Learning infinite RBMs with frank-wolfe. In *Advances in Neural Information Processing Systems*, 2016.

G Pisier. Remarques sur un résultat non publié de B. Maurey. *Séminaire Analyse fonctionnelle (dit" Maurey-Schwartz")*, pages 1–12, 1981.

Boris T Polyak. Gradient methods for the minimisation of functionals. *USSR Computational Mathematics and Mathematical Physics*, 3(4):864–878, 1963.

Boris Teodorovich Polyak. Existence theorems and convergence of minimizing sequences for extremal problems with constraints. In *Doklady Akademii Nauk*, volume 166, pages 287–290. Russian Academy of Sciences, 1966.

Mihai Pop. Shotgun sequence assembly. *Advances in computers*, 60:193–248, 2004.

Huaijun Qiu and Edwin R Hancock. Clustering and embedding using commute times. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11), 2007.

Chao Qu, Yan Li, and Huan Xu. Non-convex conditional gradient sliding. *arXiv preprint arXiv:1708.04783*, 2017.

Nikhil Rao, Parikshit Shah, and Stephen Wright. Forward–backward greedy algorithms for atomic norm regularization. *IEEE Transactions on Signal Processing*, 63(21):5798–5811, 2015.

Antoine Recanati, Thomas Brüls, and Alexandre d'Aspremont. A spectral algorithm for fast de novo layout of uncorrected long nanopore reads. *Bioinformatics*, 2016.

Antoine Recanati, Thomas Kerdreux, and Alexandre d'Aspremont. Reconstructing latent orderings by spectral clustering. *arXiv preprint arXiv:1807.07122*, 2018a.

Antoine Recanati, Nicolas Servant, Jean-Philippe Vert, and Alexandre d'Aspremont. Robust seriation and applications to cancer genomics. *arXiv preprint arXiv:1806.00664*, 2018b.

Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alex Smola. Stochastic frank-wolfe methods for nonconvex optimization. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1244–1251. IEEE, 2016.

Lothar Reichel and Lloyd N Trefethen. Eigenvalues and pseudo-eigenvalues of toeplitz matrices. *Linear algebra and its applications*, 162:153–185, 1992.

Gerhard Reinelt. *The traveling salesman: computational solutions for TSP applications*. Springer-Verlag, 1994.

Emile Richard, Guillaume R Obozinski, and Jean-Philippe Vert. Tight convex relaxations for sparse matrix factorization. In *Advances in neural information processing systems*, pages 3284–3292, 2014.

Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 2014.

Stephen M Robinson. An application of error bounds for convex programming in a linear space. *SIAM Journal on Control*, 13(2):271–273, 1975.

Stephen M Robinson. Generalized equations and their solutions, part ii: Applications to nonlinear programming. In *Optimality and Stability in Mathematical Programming*, pages 200–221. Springer, 1982.

R. T. Rockafellar. *Convex Analysis*. Princeton University Press., Princeton., 1970a.

R. Tyrrell Rockafellar. Convex analysis. In *Princeton Landmarks in Mathematics and Physics*, 1970b.

Vincent Roulet. *On the geometry of optimization problems and their structure.* PhD thesis, Paris Sciences et Lettres, 2017.

Vincent Roulet and Alexandre d'Aspremont. Sharpness, restart and acceleration. In *Advances in Neural Information Processing Systems*, pages 1119–1129, 2017.

Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

Simone Scardapane, Danilo Comminiello, Amir Hussain, and Aurelio Uncini. Group sparse regularization for deep neural networks. *Neurocomputing*, 241:81–89, 2017.

Markus Schneider. Probability inequalities for kernel embeddings in sampling without replacement. In *Artificial Intelligence and Statistics*, pages 66–74, 2016.

Rolf Schneider. *Convex bodies: the Brunn–Minkowski theory.* Number 151. Cambridge university press, 2014.

Rolf Schneider. Curvatures of typical convex bodies—the complete picture. *Proceedings of the American Mathematical Society*, 143(1):387–393, 2015.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, pages 583–588. Springer, 1997.

Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2003.

Guillaume Seguin, Piotr Bojanowski, Rémi Lajugie, and Ivan Laptev. Instance-level video segmentation from object tracks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3678–3687, 2016.

Robert J Serfling. Probability inequalities for the sum in sampling without replacement. *The Annals of Statistics*, pages 39–48, 1974.

Shai Shalev-Shwartz. *Online Learning: Theory, Algorithms, and Applications.* PhD thesis, 2007.

Shai Shalev-Shwartz, Alon Gonen, and Ohad Shamir. Large-scale convex minimization with a low-rank constraint. *arXiv preprint arXiv:1106.1622*, 2011.

Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

B Simon et al. *Trace ideals and their applications.* American Mathematical Soc., 1979.

Amit Singer. From graph to manifold laplacian: The convergence rate. *Applied and Computational Harmonic Analysis*, 21(1):128–134, 2006.

W So. Facial structures of schatten p-norms. *Linear and Multilinear Algebra*, 27(3):207–212, 1990.

Karthik Sridharan. A gentle introduction to concentration inequalities. 2002.

Weijie Su, Stephen Boyd, and Emmanuel J Candes. A differential equation for modeling nesterov's accelerated gradient method: theory and insights. *The Journal of Machine Learning Research*, 17 (1):5312–5354, 2016.

Yifan Sun and Francis Bach. Safe screening for the generalized conditional gradient method. *arXiv:2002.09718*, 2020.

Vladimir Temlyakov. *Greedy approximation*, volume 20. Cambridge University Press, 2011.

Robert Tibshirani, Jacob Bien, Jerome Friedman, Trevor Hastie, Noah Simon, Jonathan Taylor, and Ryan J. Tibshirani. Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 2012.

Michael J Todd and E Alper Yıldırım. On khachiyan's algorithm for the computation of minimum-volume enclosing ellipsoids. *Discrete Applied Mathematics*, 155(13):1731–1744, 2007.

Nicole Tomczak-Jaegermann. The moduli of smoothness and convexity and the rademacher averages of the trace classes. *Studia Mathematica*, 50(2):163–182, 1974.

Ryota Tomioka and Taiji Suzuki. Convex tensor decomposition via structured schatten norm regularization. In *Advances in neural information processing systems*, pages 1331–1339, 2013.

William F Trench. On the eigenvalue problem for toeplitz band matrices. *Linear Algebra and its Applications*, 64:199–214, 1985.

Titouan Vayer, Laetitia Chapel, Rémi Flamary, Romain Tavenard, and Nicolas Courty. Optimal transport for structured data with application on graphs. *arXiv preprint arXiv:1805.09114*, 2018.

Roman Vershynin. How close is the sample covariance matrix to the actual covariance matrix? *Journal of Theoretical Probability*, 25(3):655–686, 2012.

Jean-Philippe Vial. Strong and weak convexity of sets and functions. *Mathematics of Operations Research*, 8(2):231–259, 1983.

Marina Vinyes and Guillaume Obozinski. Fast column generation for atomic norm regularization. In *The 20th International Conference on Artificial Intelligence and Statistics*, 2017.

Joshua T Vogelstein, John M Conroy, Vince Lyzinski, Louis J Podrazik, Steven G Kratzer, Eric T Harley, Donniell E Fishkind, R Jacob Vogelstein, and Carey E Priebe. Fast approximate quadratic programming for graph matching. *PLOS one*, 2015.

Balder Von Hohenbalken. Simplicial decomposition in nonlinear programming algorithms. *Mathematical Programming*, 13(1):49–68, 1977.

U. Von Luxburg, O. Bousquet, and M. Belkin. Limits of spectral clustering. In *Advances in Neural Information Processing Systems 17*, pages 857–864, Cambridge, MA, USA, July 2005. Max-Planck-Gesellschaft, MIT Press.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

Ha Huy Vui. Global holderian error bound for nondegenerate polynomials. *SIAM Journal on Optimization*, 23(2):917–933, 2013.

Po-Wei Wang and Chih-Jen Lin. Iteration complexity of feasible descent methods for convex optimization. *Journal of Machine Learning Research*, 2014.

Alexander Weber and Gunther Reisig. Local characterization of strongly convex sets. *Journal of Mathematical Analysis and Applications*, 400(2):743–750, 2013.

Kishan Wimalawarne, Masashi Sugiyama, and Ryota Tomioka. Multitask learning meets tensor factorization: task imputation via convex optimization. In *Advances in neural information processing systems*, pages 2825–2833, 2014.

Philip Wolfe. Convergence theory in nonlinear programming. *In Integer and Nonlinear Programming*, 1970.

Philip Wolfe. Finding the nearest point in a polytope. *Mathematical Programming*, 11(1):128–149, 1976.

Yi Xu and Tianbao Yang. Frank-wolfe method is automatically adaptive to error bound condition, 2018.

Peiran Yu, Guoyin Li, and Ting Kei Pong. Deducing kurdyka-lojasiewicz exponent via inf-projection. *arXiv preprint arXiv:1902.03635*, 2019.

Yaoliang Yu, Xinhua Zhang, and Dale Schuurmans. Generalized conditional gradient for sparse estimation. *The Journal of Machine Learning Research*, 18(1):5279–5324, 2017.

Yi Yu, Tengyao Wang, and Richard J Samworth. A useful variant of the davis–kahan theorem for statisticians. *Biometrika*, 102(2):315–323, 2014.

Alp Yurtsever, Madeleine Udell, Joel A Tropp, and Volkan Cevher. Sketchy decisions: Convex low-rank matrix optimization with optimal storage. *arXiv preprint arXiv:1702.06838*, 2017.

Mikhail Zaslavskiy, Francis Bach, and Jean-Philippe Vert. A path following algorithm for the graph matching problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

Mingrui Zhang, Zebang Shen, Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. One sample stochastic frank-wolfe. *arXiv preprint arXiv:1910.04322*, 2019.

Denny Zhou, Jason Weston, Arthur Gretton, Olivier Bousquet, and Bernhard Schölkopf. Ranking on data manifolds. In *Advances in neural information processing systems*, pages 169–176, 2004.

Zirui Zhou and Anthony Man-Cho So. A unified approach to error bounds for structured convex optimization problems. *Mathematical Programming*, 165(2):689–728, 2017.

RÉSUMÉ

Les algorithmes de Frank-Wolfe sont des méthodes d'optimisation de problèmes sous contraintes. Elles décomposent un problème non-linéaire en une série de problèmes linéaires. Cela en fait des méthodes de choix pour l'optimisation en grande dimension et notamment explique leur utilisation dans de nombreux domaines appliqués. Ici nous proposons de nouveaux algorithmes de Frank-Wolfe qui convergent plus rapidement vers la solution du problème d'optimisation sous certaines hypothèses structurelles assez génériques. Nous montrons en particulier, que contrairement à d'autres types d'algorithmes, cette famille s'adapte à ces hypothèses sans avoir à spécifier les paramètres qui les contrôlent.

MOTS CLÉS

Frank-Wolfe, Gradient Conditionnel, Inégalité de Łojasiewciz, Convexité Uniforme, Carathéodory Approximé, Taux de convergence

ABSTRACT

The Frank-Wolfe algorithms, a.k.a. conditional gradient algorithms, solve constrained optimization problems. They break down a non-linear problem into a series of linear minimization on the constraint set. This contributes to their recent revival in many applied domains, in particular those involving large-scale optimization problems. In this dissertation, we design or analyze versions of the Frank-Wolfe algorithms. We notably show that, contrary to other types of algorithms, this family is adaptive to a broad spectrum of structural assumptions, without the need to know and specify the parameters controlling these hypotheses.

KEYWORDS

Frank-Wolfe, Conditional Gradient, Łojasiweciz Inequality, Uniform Convexity, Approximate Carathéodory, convergence rates