Conditional Gradients: a tour d'horizon

Sebastian Pokutta

Technische Universität Berlin
and
Zuse Institute Berlin

pokutta@math.tu-berlin.de
@spokutta

Tech Lunch Talk · July 30, 2021

Berlin Mathematics Research Center
MATH+

What is this talk about?

Introduction

A very versatile and simple optimization method for projection-free optimization that promotes sparsity.

What is this talk about?

Introduction

A very versatile and simple optimization method for projection-free optimization that promotes sparsity.

Why? Constraints and Sparsity help interpretability and explainability.

What is this talk about?

Introduction

A very versatile and simple optimization method for projection-free optimization that promotes sparsity.

Why? Constraints and Sparsity help interpretability and explainability.

Today: A brief overview of recent developments in *conditional gradient methods*.

What is this talk about?

Introduction

A very versatile and simple optimization method for projection-free optimization that promotes sparsity.

Why? Constraints and Sparsity help interpretability and explainability.

Today: A brief overview of recent developments in *conditional gradient methods*.

Outline

- The basics: Conditional Gradients a.k.a. the Frank-Wolfe algorithm
- Several examples:
  - Deep Learning
  - Rate-Distortion Explanation
  - Sparse Regression
  - Matrix Completion
- High-performance Julia Package: FrankWolfe.jl

(Hyperlinked) References are not exhaustive; check references contained therein.

Conditional Gradients
a.k.a. the Frank-Wolfe algorithm

—The Basics—

The basic problem

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

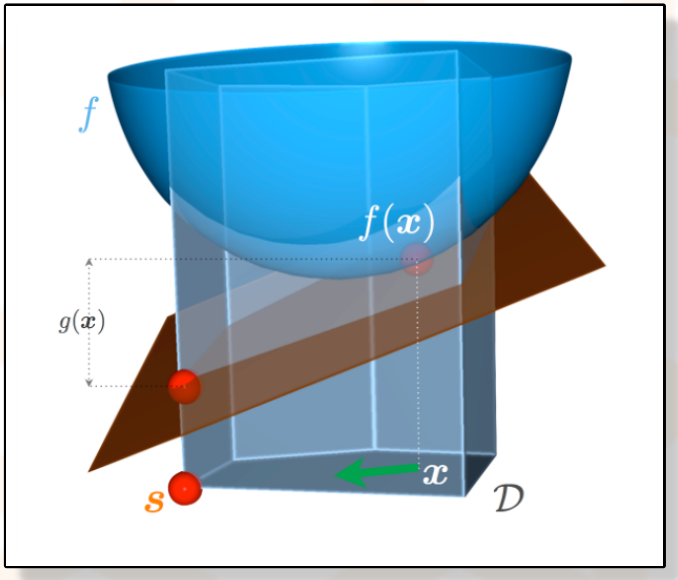
Given a smooth and convex function f and a polytope P, solve optimization problem:

The basic problem

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Given a smooth and convex function f and a polytope P, solve optimization problem:

$$\min_{x \in P} f(x) \qquad \text{(baseProblem)}$$



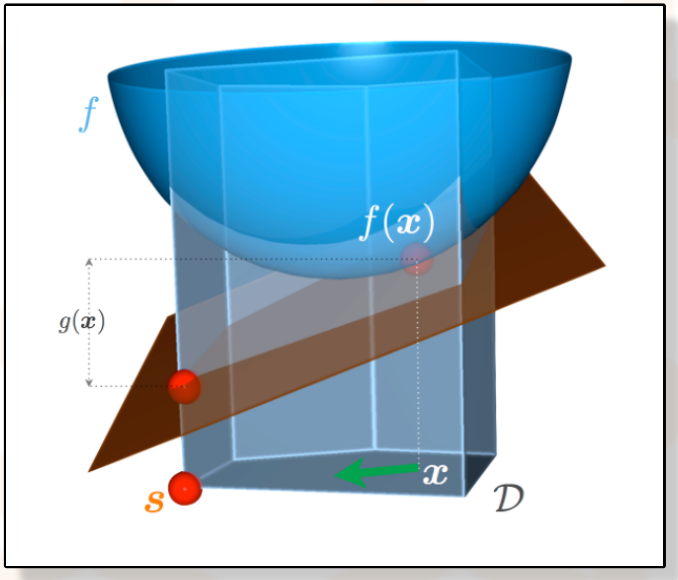
Source: [Jaggi, 2013]

The basic problem

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Given a smooth and convex function f and a polytope P, solve optimization problem:

$$\min_{x \in P} f(x) \qquad \text{(baseProblem)}$$



Source: [Jaggi, 2013]

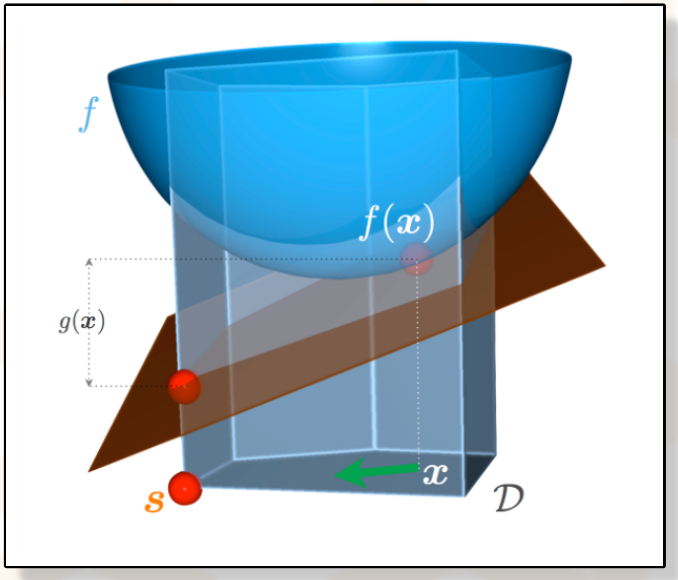
1. Very versatile model
2. Can use various types of information about both f and P
3. Works very well in (continuous) real-world applications
4. At the core of many (all?) learning algorithms (albeit mostly non-convex case)

The basic problem

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Given a smooth and convex function f and a polytope P, solve optimization problem:

$$\min_{x \in P} f(x) \qquad \text{(baseProblem)}$$



Source: [Jaggi, 2013]

Our setup.

1. Access to P. Linear Minimization Oracle (LMO): Given linear objective c return

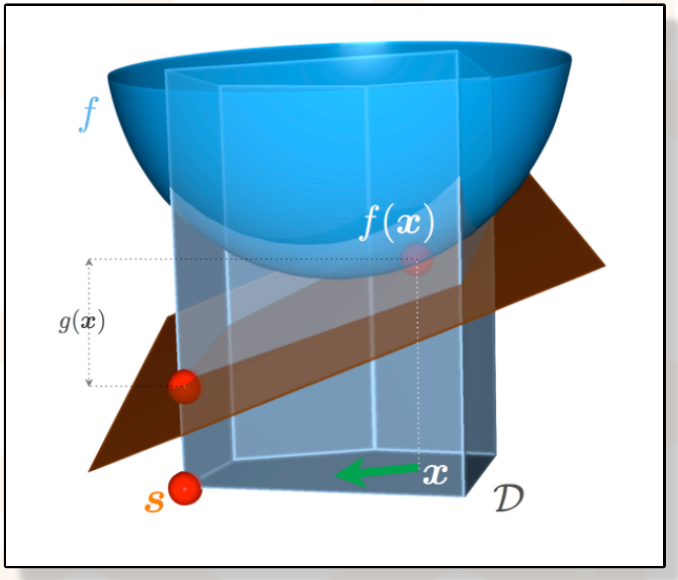
$$x \leftarrow \arg\min_{v \in P} c^T v.$$

The basic problem

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Given a smooth and convex function f and a polytope P, solve optimization problem:

$$\min_{x \in P} f(x) \qquad \text{(baseProblem)}$$



Source: [Jaggi, 2013]

Our setup.

1. Access to P. Linear Minimization Oracle (LMO): Given linear objective c return

$$x \leftarrow \arg\min_{v \in P} c^T v.$$

2. Access to f. First-Order Oracle (FO): Given x return

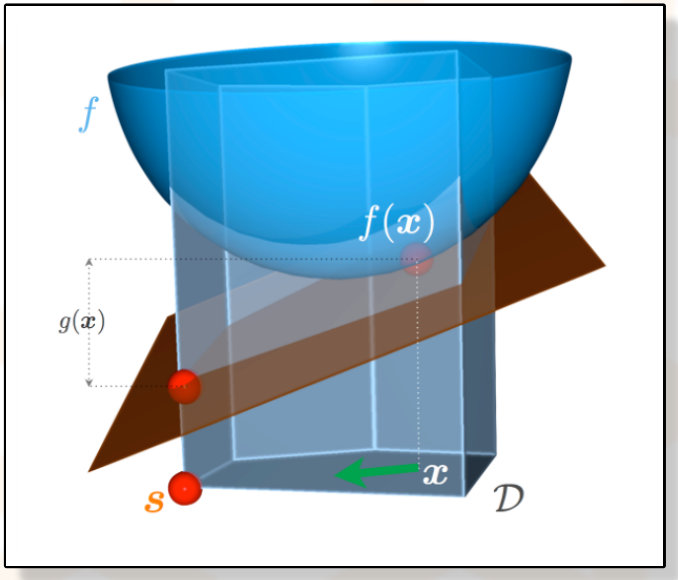
$$\nabla f(x) \qquad \text{and} \qquad f(x).$$

The basic problem

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Given a smooth and convex function f and a polytope P, solve optimization problem:

$$\min_{x \in P} f(x) \qquad \text{(baseProblem)}$$



Source: [Jaggi, 2013]

Our setup.

1. Access to P. Linear Minimization Oracle (LMO): Given linear objective c return
$$x \leftarrow \arg\min_{v \in P} c^T v.$$
2. Access to f. First-Order Oracle (FO): Given x return
$$\nabla f(x) \qquad \text{and} \qquad f(x).$$

$\Rightarrow$ *Complexity of convex optimization relative to LO/FO oracle*

Interlude: why LMOs?

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

LMO model has many advantages.

1. Includes explicit formulation via constraints
2. Some problems do not posess 'small' formulations but have efficient LMOs.
   Example: Matching Polytope [Rothvoss, 2014, Braun and Pokutta, 2015a,b, Braun et al., 2015, 2017b]
3. Allows modeling of compact convex constraints as long as we have an LMO.
   Example: SDP cone
4. Often much faster than projection.
   Example: nuclear norm. Largest singular vector (Lanczos method) vs. full SVD
5. LMO is a black box for the algorithms
6. For many LMOs of interest close form solutions available.
   Example: ℓ_1-ball for LASSO regression.

For an overview see: [Combettes and Pokutta, 2021+]

The basic problem

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Basic notions. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function.

Basic notions. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function.

Definition (Convexity)

For all x, y it holds:

$$f(y) - f(x) \geq \langle \nabla f(x), y - x \rangle .$$

In particular, all local minima are global minima.

The basic problem

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Basic notions. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function.

Definition (Convexity)

For all x, y it holds:

$$f(y) - f(x) \geq \langle \nabla f(x), y - x \rangle .$$

In particular, all local minima are global minima.

Definition (L-Smoothness)

For all x, y it holds:

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

The basic problem

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

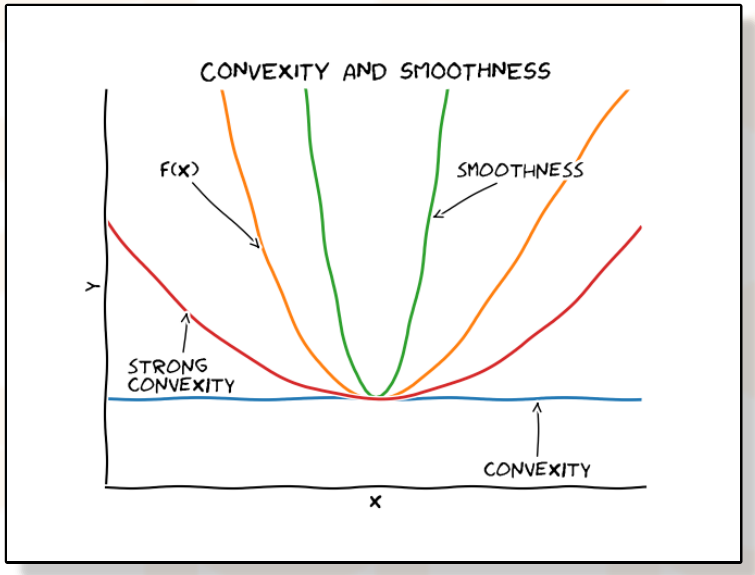
Basic notions. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function.

Definition (Convexity)

For all x, y it holds:

$$f(y) - f(x) \geq \langle \nabla f(x), y - x \rangle .$$

In particular, all local minima are global minima.



Definition (L-Smoothness)

For all x, y it holds:

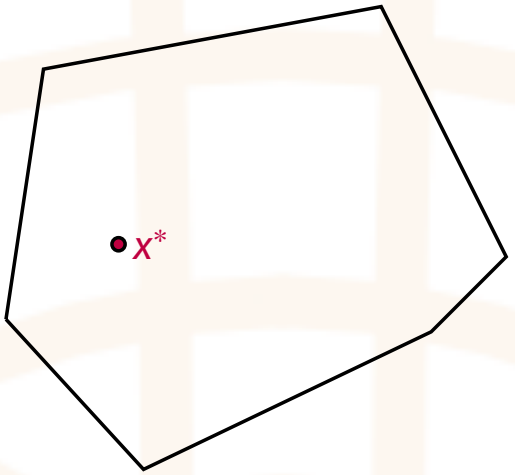
$$f(y) - f(x) \leq \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2 .$$

The Frank-Wolfe Algorithm

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

1: $x_0 \in P$
2: **for** $t = 0$ **to** $T - 1$ **do**
3: $\quad v_t \leftarrow \arg\min_{v \in P} \langle \nabla f(x_t), v \rangle$
4: $\quad x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$
5: **end for**



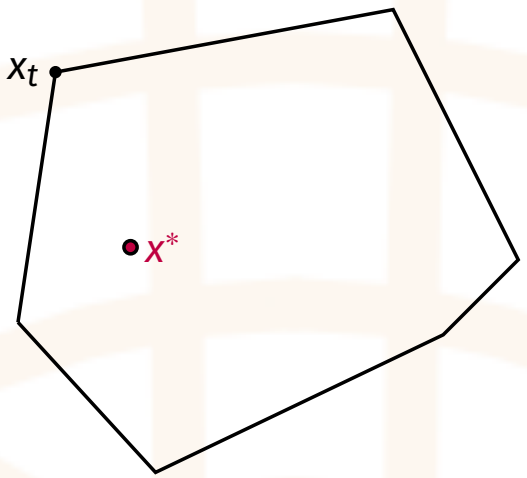
[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

The Frank-Wolfe Algorithm

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

1: $x_0 \in P$
2: **for** $t = 0$ **to** $T - 1$ **do**
3: $\quad v_t \leftarrow \arg\min_{v \in P} \langle \nabla f(x_t), v \rangle$
4: $\quad x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$
5: **end for**



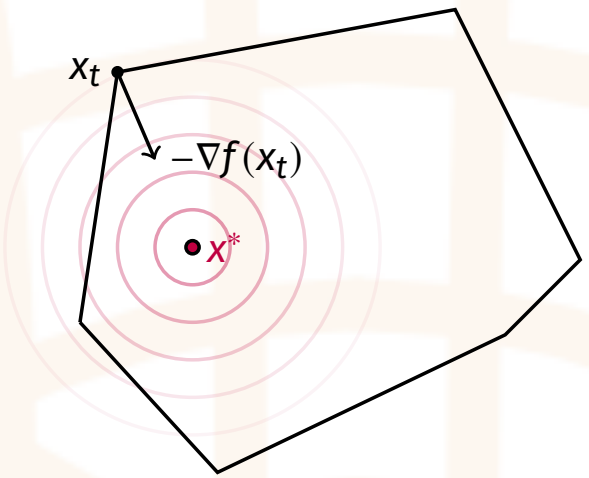
[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

The Frank-Wolfe Algorithm

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

1: $x_0 \in P$
2: **for** $t = 0$ **to** $T - 1$ **do**
3: $\quad v_t \leftarrow \arg\min_{v \in P} \langle \nabla f(x_t), v \rangle$
4: $\quad x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$
5: **end for**



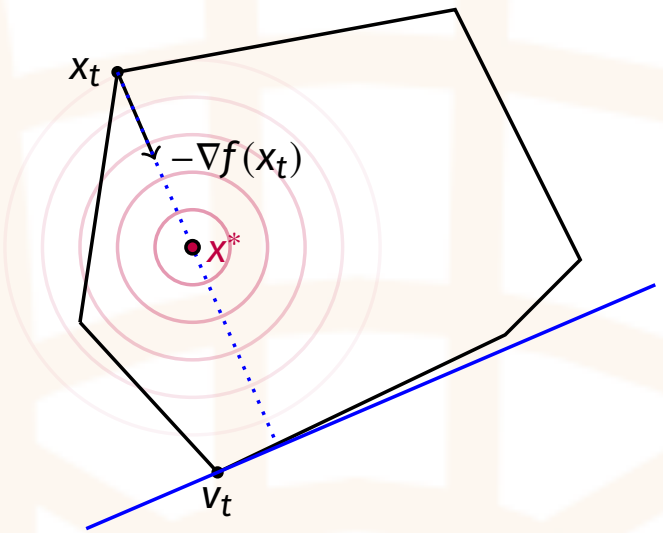
[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

The Frank-Wolfe Algorithm

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

1: $x_0 \in P$
2: **for** $t = 0$ **to** $T - 1$ **do**
3: $\quad v_t \leftarrow \arg\min_{v \in P} \langle \nabla f(x_t), v \rangle$
4: $\quad x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$
5: **end for**



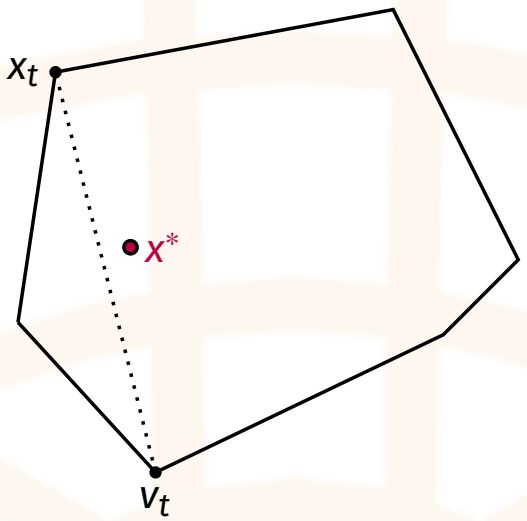
[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

The Frank-Wolfe Algorithm

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

1: $x_0 \in P$
2: **for** $t = 0$ **to** $T - 1$ **do**
3: $\quad v_t \leftarrow \arg\min_{v \in P} \langle \nabla f(x_t), v \rangle$
4: $\quad x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$
5: **end for**



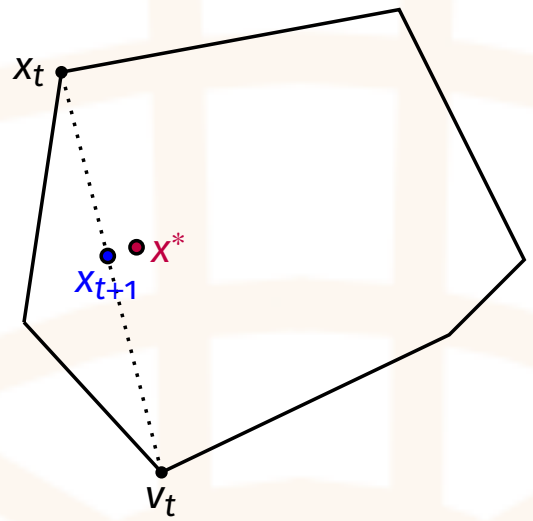
[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

The Frank-Wolfe Algorithm

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

1: $x_0 \in P$
2: **for** $t = 0$ **to** $T - 1$ **do**
3: $\quad v_t \leftarrow \arg\min_{v \in P} \langle \nabla f(x_t), v \rangle$
4: $\quad x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$
5: **end for**



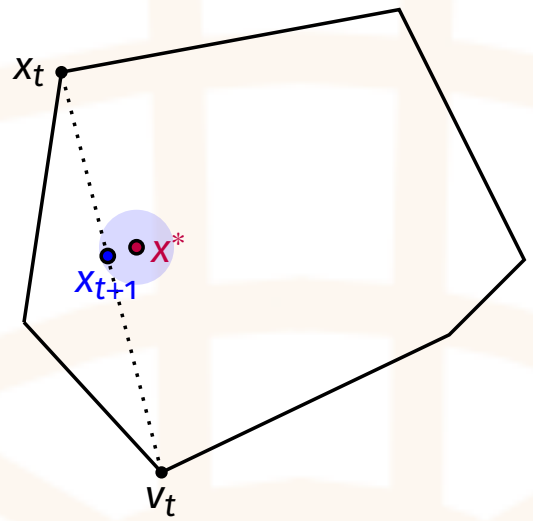
[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

The Frank-Wolfe Algorithm

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

1: $x_0 \in P$
2: **for** $t = 0$ **to** $T - 1$ **do**
3: $\quad v_t \leftarrow \arg\min_{v \in P} \langle \nabla f(x_t), v \rangle$
4: $\quad x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$
5: **end for**



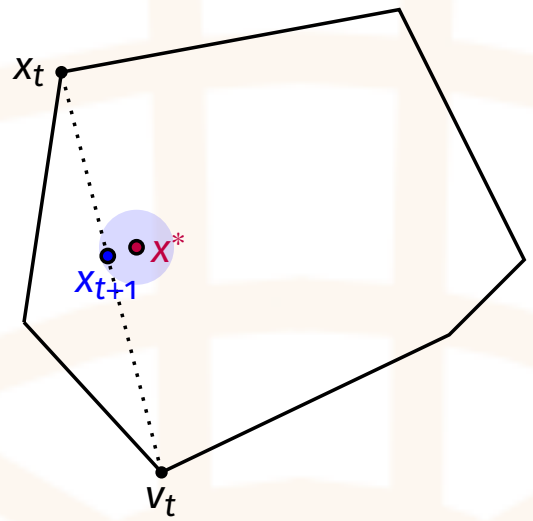
[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

The Frank-Wolfe Algorithm

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

1: $x_0 \in P$
2: **for** $t = 0$ **to** $T - 1$ **do**
3: $\quad v_t \leftarrow \arg\min_{v \in P} \langle \nabla f(x_t), v \rangle$
4: $\quad x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$
5: **end for**



[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

Advantages:

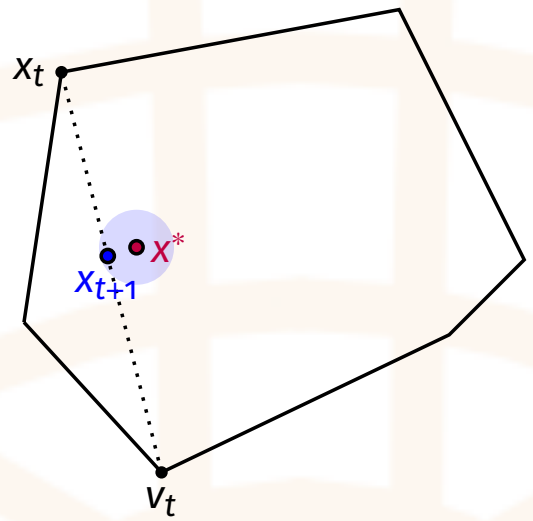
- Extremely simple and robust: no complicated data structures to maintain
- Easy to implement: requires only the two oracles
- Projection-free: feasibility convex combination and LO oracle.
- Sparsity: optimal solution is a convex combination of (usually) vertices.

The Frank-Wolfe Algorithm

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

1: $x_0 \in P$
2: **for** $t = 0$ **to** $T - 1$ **do**
3: $\quad v_t \leftarrow \arg\min_{v \in P} \langle \nabla f(x_t), v \rangle$
4: $\quad x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$
5: **end for**



[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

Advantages:

- Extremely simple and robust: no complicated data structures to maintain
- Easy to implement: requires only the two oracles
- Projection-free: feasibility convex combination and LO oracle.
- Sparsity: optimal solution is a convex combination of (usually) vertices.

Disadvantages:

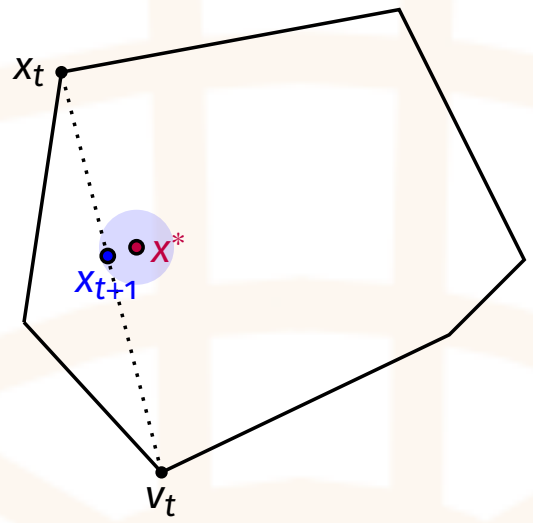
- Suboptimal convergence rate of $O(1/T)$

The Frank-Wolfe Algorithm

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

1: $x_0 \in P$
2: **for** $t = 0$ **to** $T - 1$ **do**
3: $\quad v_t \leftarrow \arg\min_{v \in P} \langle \nabla f(x_t), v \rangle$
4: $\quad x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$
5: **end for**



[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

Advantages:

- Extremely simple and robust: no complicated data structures to maintain
- Easy to implement: requires only the two oracles
- Projection-free: feasibility convex combination and LO oracle.
- Sparsity: optimal solution is a convex combination of (usually) vertices.

Disadvantages:

- Suboptimal convergence rate of $O(1/T)$

$\Rightarrow$ *Despite (theoretically) suboptimal rate heavily used in applications due to simplicity.*

Simple Convergence Proof

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Theorem (Convergence rate of the vanilla Frank-Wolfe Algorithm)

Let f be L-smooth convex, P be polytope with diameter D. With choice $\gamma_t \doteq \frac{2}{t+3}$:

$$f(x_t) - f(x^*) \leq \frac{2LD^2}{t+3}.$$

Simple Convergence Proof

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Theorem (Convergence rate of the vanilla Frank-Wolfe Algorithm)

Let f be L-smooth convex, P be polytope with diameter D. With choice $\gamma_t \doteq \frac{2}{t+3}$:

$$f(x_t) - f(x^*) \leq \frac{2LD^2}{t+3}.$$

Proof Sketch.

By smoothness:

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

Simple Convergence Proof

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Theorem (Convergence rate of the vanilla Frank-Wolfe Algorithm)

Let f be L-smooth convex, P be polytope with diameter D. With choice $\gamma_t \doteq \frac{2}{t+3}$:

$$f(x_t) - f(x^*) \leq \frac{2LD^2}{t+3}.$$

Proof Sketch.

By smoothness:

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

LP maximality and convexity: $\langle \nabla f(x_t), v_t - x_t \rangle \leq \langle \nabla f(x_t), x^* - x_t \rangle \leq f(x^*) - f(x_t)$. Moreover, $\|v_t - x_t\| \leq D$.

Simple Convergence Proof

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Theorem (Convergence rate of the vanilla Frank-Wolfe Algorithm)

Let f be L-smooth convex, P be polytope with diameter D. With choice $\gamma_t \doteq \frac{2}{t+3}$:

$$f(x_t) - f(x^*) \leq \frac{2LD^2}{t+3}.$$

Proof Sketch.

By smoothness:

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

LP maximality and convexity: $\langle \nabla f(x_t), v_t - x_t \rangle \leq \langle \nabla f(x_t), x^* - x_t \rangle \leq f(x^*) - f(x_t)$. Moreover, $\|v_t - x_t\| \leq D$.

Thus:

$$f(x_{t+1}) - f(x^*) \leq (1 - \gamma_t)(f(x_t) - f(x^*)) + \gamma_t^2 \frac{LD^2}{2}.$$

□

Simple Convergence Proof

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Theorem (Convergence rate of the vanilla Frank-Wolfe Algorithm)

Let f be L-smooth convex, P be polytope with diameter D. With choice $\gamma_t \doteq \frac{2}{t+3}$:

$$f(x_t) - f(x^*) \leq \frac{2LD^2}{t+3}.$$

Proof Sketch.

By smoothness:

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

LP maximality and convexity: $\langle \nabla f(x_t), v_t - x_t \rangle \leq \langle \nabla f(x_t), x^* - x_t \rangle \leq f(x^*) - f(x_t)$. Moreover, $\|v_t - x_t\| \leq D$.

Thus:

$$f(x_{t+1}) - f(x^*) \leq (1 - \gamma_t)(f(x_t) - f(x^*)) + \gamma_t^2 \frac{LD^2}{2}.$$

By Induction (plugging in the guarantee + definition of γ_t):

$$f(x_{t+1}) - f(x^*) \leq \left(1 - \frac{2}{t+3}\right) \frac{2LD^2}{t+3} + \frac{4}{(t+3)^2} \cdot \frac{LD^2}{2} = \frac{2LD^2(t+2)}{(t+3)^2} \leq \frac{2LD^2}{t+4},$$

by $(t+2)(t+4) \leq (t+3)^2$.

□

A matching lower bound

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Consider $P = \operatorname{conv}(\{e_1, \dots, e_n\})$ the probability simplex and $f = \|x\|^2$.

A matching lower bound

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Consider $P = \operatorname{conv}(\{e_1, \dots, e_n\})$ the probability simplex and $f = \|x\|^2$.

Clearly. $\arg\min_{x \in P} f(x) = x^* \doteq \frac{1}{n} e$ with $f(x^*) = \frac{1}{n}$.

A matching lower bound

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Consider $P = \operatorname{conv}(\{e_1, \dots, e_n\})$ the probability simplex and $f = \|x\|^2$.

Clearly. $\arg\min_{x \in P} f(x) = x^* \doteq \frac{1}{n} e$ with $f(x^*) = \frac{1}{n}$.

Observe. Starting from any vertex e_i after $t < n$ iterations we picked up at most t vertices of P.

A matching lower bound

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Consider $P = \operatorname{conv}(\{e_1, \dots, e_n\})$ the probability simplex and $f = \|x\|^2$.

Clearly. $\arg\min_{x \in P} f(x) = x^* \doteq \frac{1}{n} e$ with $f(x^*) = \frac{1}{n}$.

Observe. Starting from any vertex e_i after $t < n$ iterations we picked up at most t vertices of P.

Easy to see. For any iterate x_t:

$$f(x_t) \geq \min_{\substack{x \in \operatorname{conv}(\mathcal{S}) \\ \mathcal{S} \subseteq \{e_1, \dots, e_n\} \\ |\mathcal{S}| \leq t}} f(x) = 1/t,$$

A matching lower bound

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Consider $P = \operatorname{conv}(\{e_1, \dots, e_n\})$ the probability simplex and $f = \|x\|^2$.

Clearly. $\arg\min_{x \in P} f(x) = x^* \doteq \frac{1}{n} e$ with $f(x^*) = \frac{1}{n}$.

Observe. Starting from any vertex e_i after $t < n$ iterations we picked up at most t vertices of P.

Easy to see. For any iterate x_t:

$$f(x_t) \geq \min_{\substack{x \in \operatorname{conv}(\mathcal{S}) \\ \mathcal{S} \subseteq \{e_1, \dots, e_n\} \\ |\mathcal{S}| \leq t}} f(x) = 1/t,$$

Thus lower bound. $f(x_t) - f(x^*) \geq \frac{1}{t} - \frac{1}{n}$

$\Rightarrow$ *Any LP method converges no faster than $O(1/t)$.*

Note: Strong consequences for strongly convex case and also provides a sparsity vs. optimality trade-off.

see also for non-smooth variants: [Braun et al., 2017a]

A matching lower bound

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Consider $P = \operatorname{conv}(\{e_1, \dots, e_n\})$ the probability simplex and $f = \|x\|^2$.

Clearly. $\arg\min_{x \in P} f(x) = x^* \doteq \frac{1}{n} e$ with $f(x^*) = \frac{1}{n}$.

Observe. Starting from any vertex e_i after $t < n$ iterations we picked up at most t vertices of P.

Easy to see. For any iterate x_t:

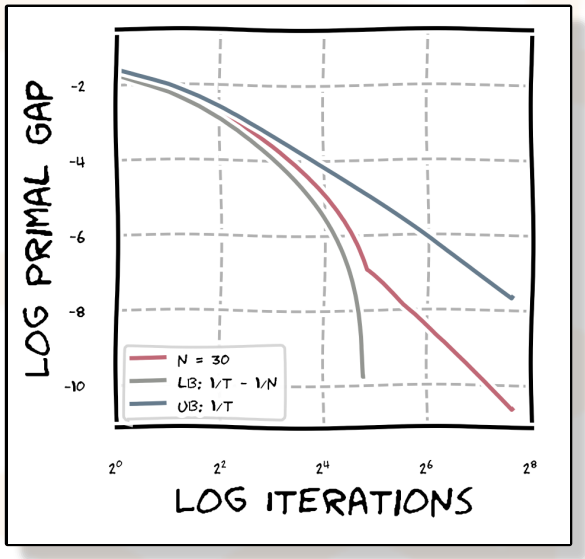
$$f(x_t) \geq \min_{\substack{x \in \operatorname{conv}(\mathcal{S}) \\ \mathcal{S} \subseteq \{e_1, \dots, e_n\} \\ |\mathcal{S}| \leq t}} f(x) = 1/t,$$

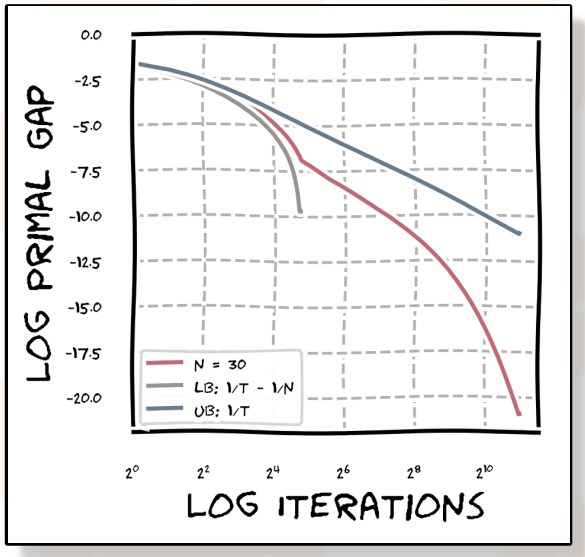
Thus lower bound. $f(x_t) - f(x^*) \geq \frac{1}{t} - \frac{1}{n}$

$\Rightarrow$ *Any LP method converges no faster than $O(1/t)$.*

Note: Strong consequences for strongly convex case and also provides a sparsity vs. optimality trade-off.

see also for non-smooth variants: [Braun et al., 2017a]





Significant progress over the recent years (incomplete list)

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

1. Strongly convex case [Garber and Hazan, 2013, Lacoste-Julien and Jaggi, 2015, Lan and Zhou, 2016, Garber and Meshi, 2016]
2. Non-convex case [Lacoste-Julien, 2016]
3. Online case [Hazan and Kale, 2012]
4. Stochastic variants and adaptive gradients [Hazan and Luo, 2016, Reddi et al., 2016, Combettes et al., 2020]
5. Sharp functions and sharp regions [Kerdreux et al., 2019, 2021a,b]
6. Acceleration [Diakonikolas et al., 2020, Bach, 2020, Carderera et al., 2021]
7. Specialized variants [Freund et al., 2017, Braun et al., 2017c, 2019b,a]

Conditional Gradients very competitive: simple, robust, real-world performance.

For more background etc see upcoming survey!

Stochastic Conditional Gradients

—Training Neural Networks with Frank-Wolfe—

joint work with Christoph Spiegel and Max Zimmer

[Pokutta et al., 2020]

The Stochastic Frank-Wolfe Algorithm (with Momentum)

Training Neural Networks with Conditional Gradients

Algorithm Stochastic FW Algorithm (SFW)

$m_0 \leftarrow 0$
for $t = 0$ **to** $T - 1$ **do**
  uniformly sample i.i.d. $i_1, \ldots, i_{b_t} \sim [\![1, m]\!]$
  $\tilde{\nabla} L(\theta_t) \leftarrow \frac{1}{b_t} \sum_{j=1}^{b_t} \nabla \ell_{i_j}(\theta_t)$
  $m_t \leftarrow (1 - \rho_t)\, m_{t-1} + \rho_t\, \tilde{\nabla} L(\theta_t)$
  $v_t \leftarrow \arg\min_{v \in P} \langle m_t, v \rangle$
  $\theta_{t+1} \leftarrow \theta_t + \alpha_t (v_t - \theta_t)$
end for

e.g., [Reddi et al., 2016]

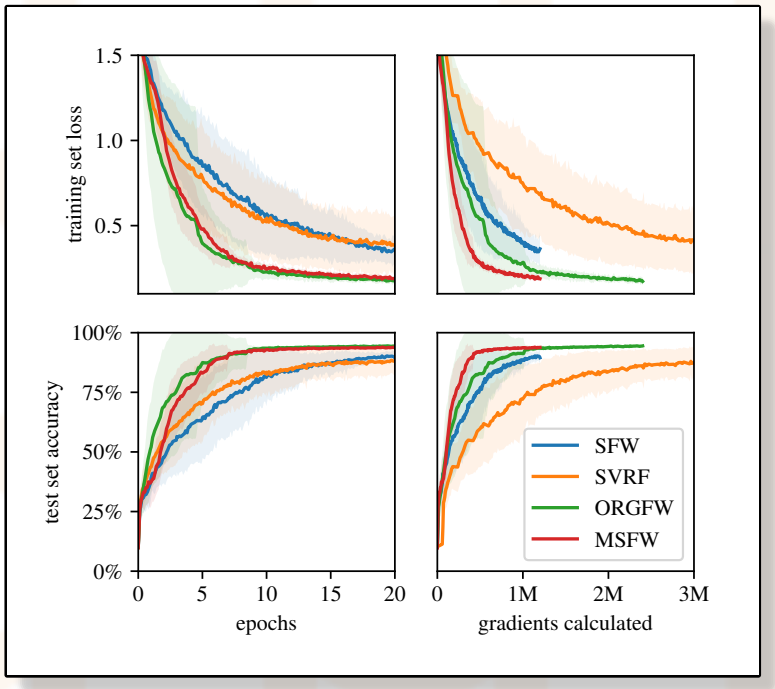
The Stochastic Frank-Wolfe Algorithm (with Momentum)

Training Neural Networks with Conditional Gradients

Algorithm Stochastic FW Algorithm (SFW)

$m_0 \leftarrow 0$
for $t = 0$ **to** $T - 1$ **do**
  uniformly sample i.i.d. $i_1, \ldots, i_{b_t} \sim [\![1, m]\!]$
  $\tilde{\nabla} L(\theta_t) \leftarrow \frac{1}{b_t} \sum_{j=1}^{b_t} \nabla \ell_{i_j}(\theta_t)$
  $m_t \leftarrow (1 - \rho_t)\, m_{t-1} + \rho_t\, \tilde{\nabla} L(\theta_t)$
  $v_t \leftarrow \arg\min_{v \in P} \langle m_t, v \rangle$
  $\theta_{t+1} \leftarrow \theta_t + \alpha_t (v_t - \theta_t)$
end for

e.g., [Reddi et al., 2016]



SFW variants

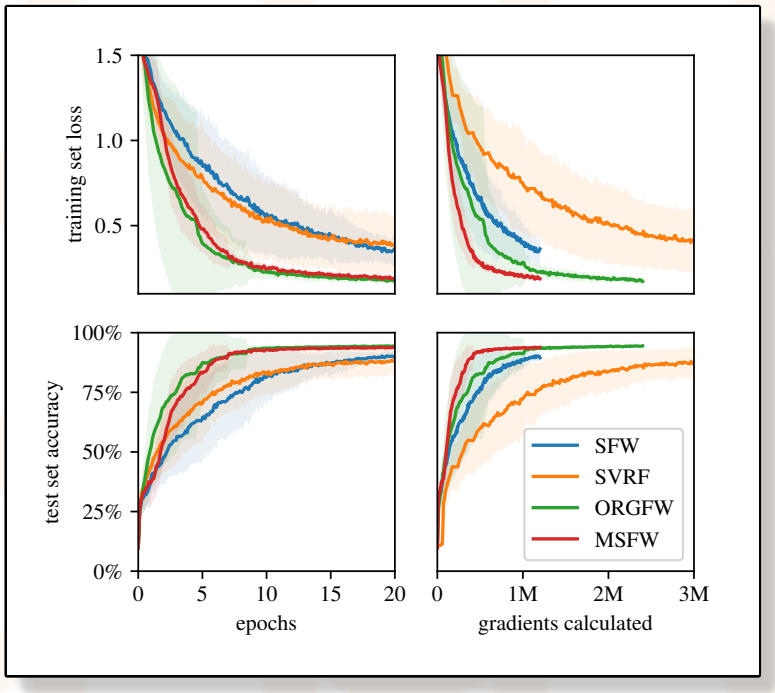
The Stochastic Frank-Wolfe Algorithm (with Momentum)

Training Neural Networks with Conditional Gradients

Algorithm Stochastic FW Algorithm (SFW)

$m_0 \leftarrow 0$
for $t = 0$ **to** $T - 1$ **do**
  uniformly sample i.i.d. $i_1, \ldots, i_{b_t} \sim [\![1, m]\!]$
  $\tilde{\nabla} L(\theta_t) \leftarrow \frac{1}{b_t} \sum_{j=1}^{b_t} \nabla \ell_{i_j}(\theta_t)$
  $m_t \leftarrow (1 - \rho_t)\, m_{t-1} + \rho_t\, \tilde{\nabla} L(\theta_t)$
  $v_t \leftarrow \arg\min_{v \in P} \langle m_t, v \rangle$
  $\theta_{t+1} \leftarrow \theta_t + \alpha_t (v_t - \theta_t)$
end for

e.g., [Reddi et al., 2016]

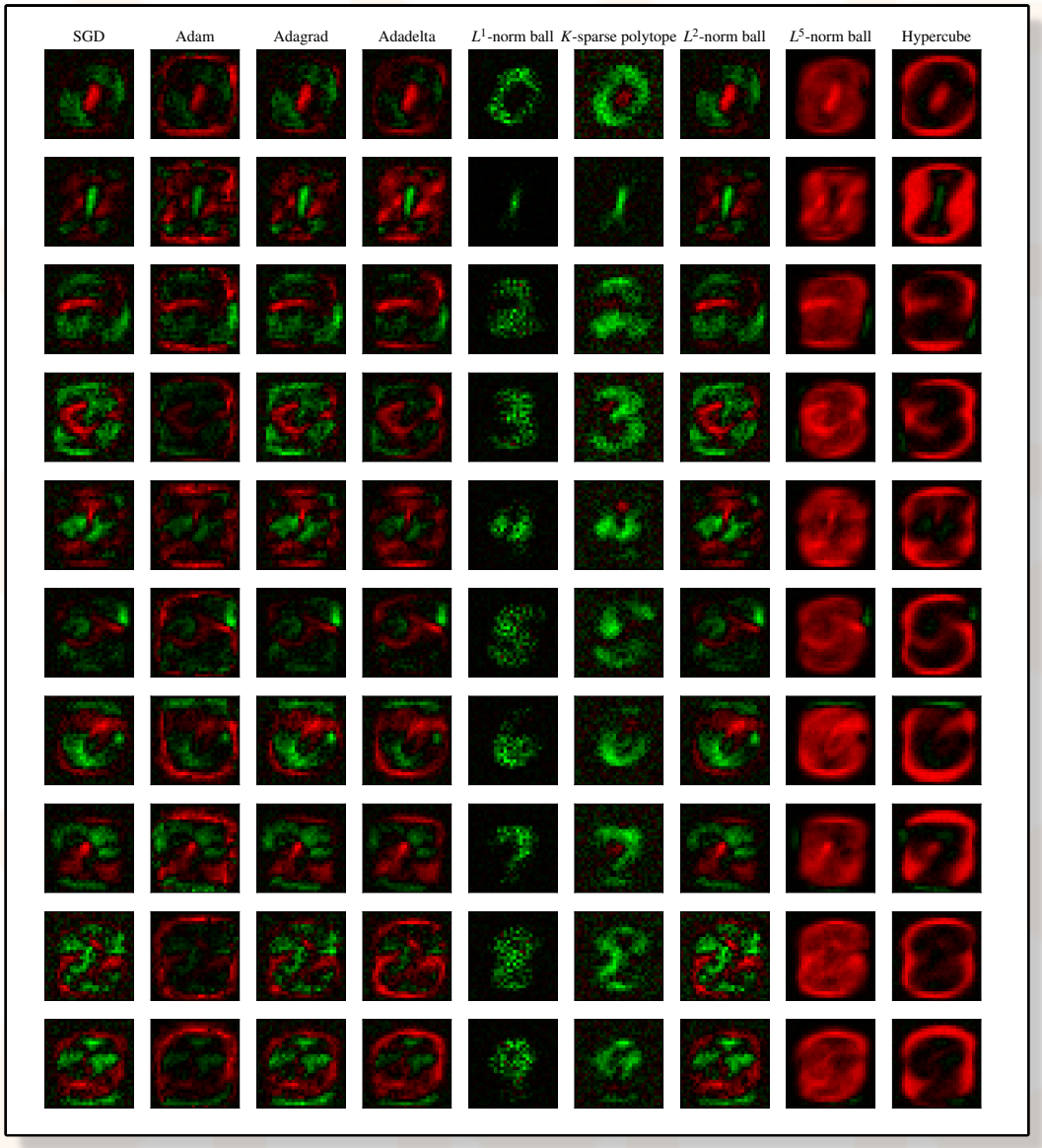


SFW variants

- Convergence rate: In the non-convex stochastic smooth case $O(1/\sqrt{T})$-rate
- Speed: Works well for very large data sets due to mini-batched gradients
- Projection-free: Remains projection-free and allows for constraints

Relevance maps under different optimizers / feasible regions

Training Neural Networks with Conditional Gradients



Robust Rate-Distortion Explanations via Conditional Gradients

joint work with Mathieu Besançon and Jan Macdonald

The problem formulation

Rate-Distortion Explanation

[Macdonald et al., 2019]

Expected Distortion of S.

$$D(S) \coloneqq D(S, \Phi, x, V) \coloneqq \underbrace{\mathbb{E}_{y \sim V}\left[\frac{1}{2}(\Phi(x) - \Phi(y))^2\right]}_{\text{Stability of } \Phi \text{ when varying outside of } S}$$

The problem formulation

Rate-Distortion Explanation

[Macdonald et al., 2019]

Expected Distortion of S.

$$D(S) \coloneqq D(S, \Phi, x, V) \coloneqq \underbrace{\mathbb{E}_{y \sim V}\left[\frac{1}{2}(\Phi(x) - \Phi(y))^2\right]}_{\text{Stability of } \Phi \text{ when varying outside of } S}$$

Rate-Distortion function.

$$R(\varepsilon) \coloneqq \underbrace{\min\{\operatorname{card}(S) : D(S) \leq \varepsilon\}}_{\text{smallest set of fixings } S}$$

The problem formulation

Rate-Distortion Explanation

[Macdonald et al., 2019]

Expected Distortion of S.

$$D(S) \coloneqq D(S, \Phi, x, V) \coloneqq \underbrace{\mathbb{E}_{y \sim V}\left[\frac{1}{2}(\Phi(x) - \Phi(y))^2\right]}_{\text{Stability of } \Phi \text{ when varying outside of } S}$$

Rate-Distortion function.

$$R(\varepsilon) \coloneqq \underbrace{\min\{\operatorname{card}(S) : D(S) \leq \varepsilon\}}_{\text{smallest set of fixings } S}$$

After convex relaxation (original problem is hard).

$$\underbrace{\min\{D(s) : \|s\|_1 \leq \lambda\}}_{\text{given budet } \lambda \text{ find } s \text{ with lowest distortion aka most relevant pixels}}$$

The problem formulation

Rate-Distortion Explanation

[Macdonald et al., 2019]

Expected Distortion of S.

$$D(S) \coloneqq D(S, \Phi, x, V) \coloneqq \underbrace{\mathbb{E}_{y \sim V}\left[\frac{1}{2}(\Phi(x) - \Phi(y))^2\right]}_{\text{Stability of } \Phi \text{ when varying outside of } S}$$

Rate-Distortion function.

$$R(\varepsilon) \coloneqq \underbrace{\min\{\operatorname{card}(S) : D(S) \leq \varepsilon\}}_{\text{smallest set of fixings } S}$$

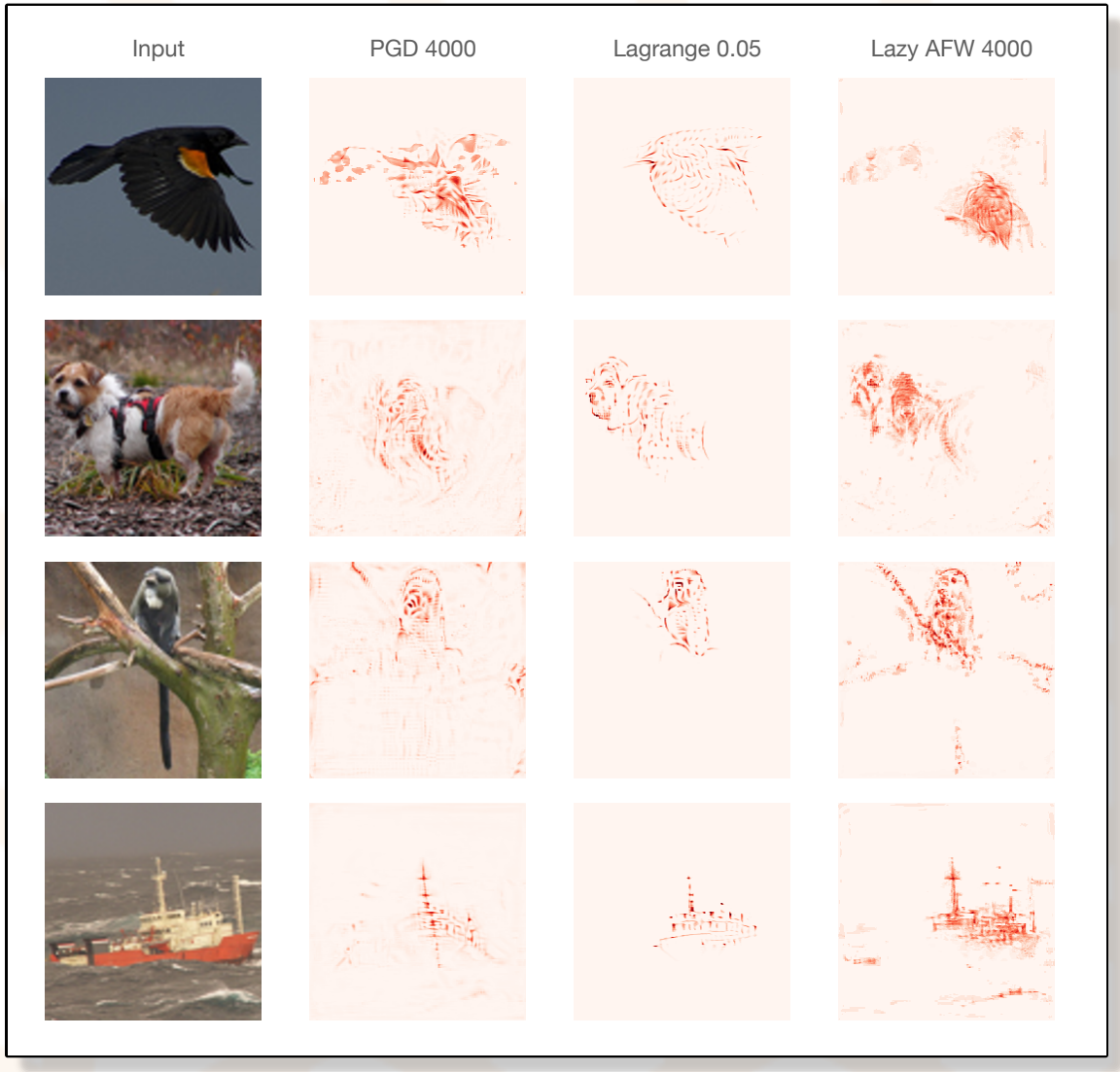
After convex relaxation (original problem is hard).

$$\underbrace{\min\{D(s) : \|s\|_1 \leq \lambda\}}_{\text{given budet } \lambda \text{ find } s \text{ with lowest distortion aka most relevant pixels}}$$

$\Rightarrow$ Typical sparse regression problem or LASSO problem.

Examples

Rate-Distortion Explanation



All methods had the same budget for picking relevant pixels. However, sparser solutions of Conditonal Gradients focus weight on most relevant pixels rather than spreading out.

Matrix Completion and Recommender Systems

The problem formulation

Matrix Completion and Recommender Systems

Task. Find matrix of low rank that coincides with given entries (observations).

Matrix Completion is a the core of many recommender systems, e.g., MovieLens.

The problem formulation

Matrix Completion and Recommender Systems

Task. Find matrix of low rank that coincides with given entries (observations).

Matrix Completion is a the core of many recommender systems, e.g., MovieLens.

Matrix Completion. Common relaxation via nuclear norm.

$$\min_{\|X\|_* \leq \tau} \sum_{(i,j) \in \mathcal{I}} \left(X_{i,j} - Y_{i,j} \right)^2$$

The problem formulation

Matrix Completion and Recommender Systems

Task. Find matrix of low rank that coincides with given entries (observations).

Matrix Completion is a the core of many recommender systems, e.g., MovieLens.

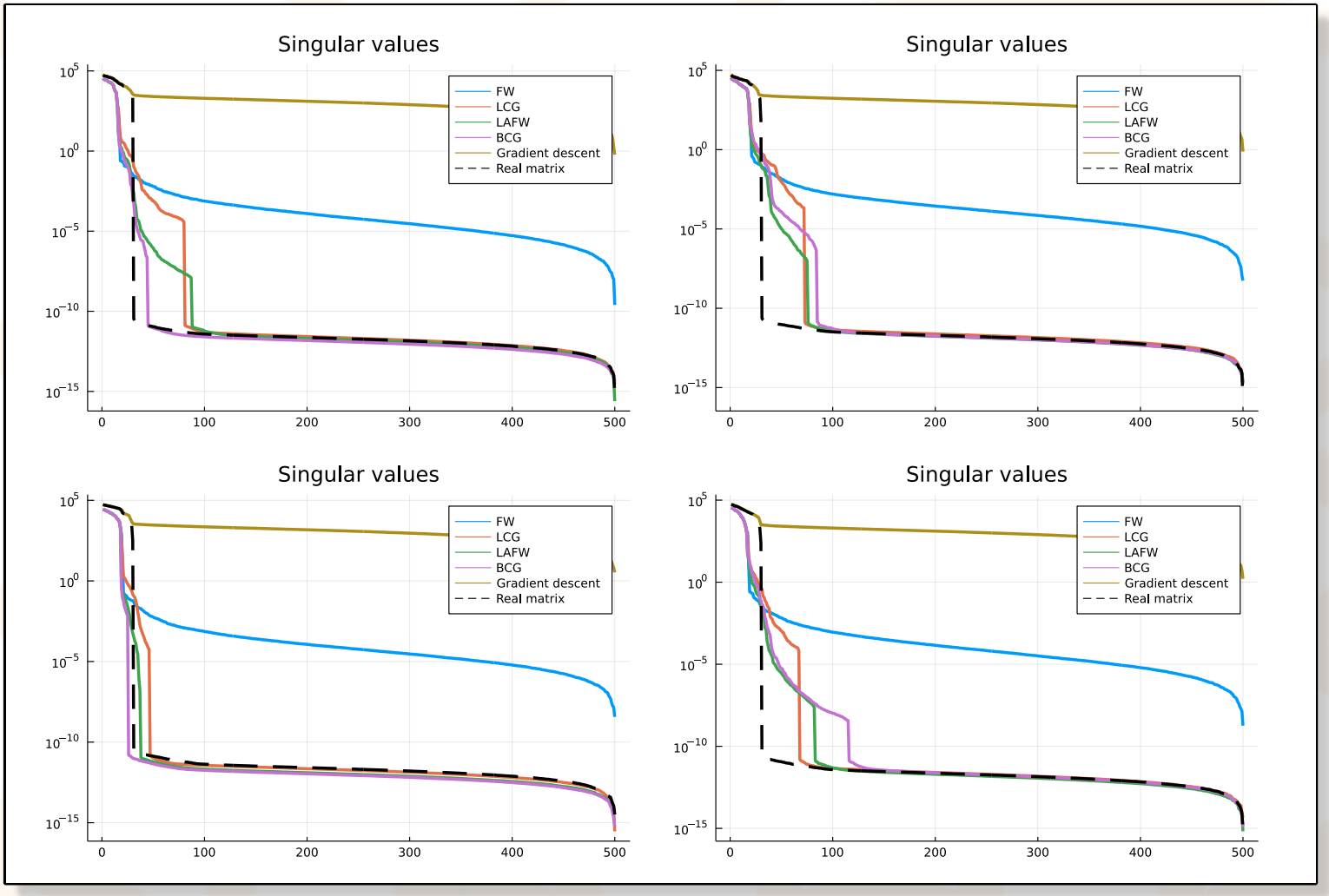
Matrix Completion. Common relaxation via nuclear norm.

$$\min_{\|X\|_* \leq \tau} \sum_{(i,j) \in \mathcal{I}} \left(X_{i,j} - Y_{i,j} \right)^2$$

Note. While nuclear norm is a convex surrogate for rank, solving the problem with, e.g., projected gradient descent does not provide low rank solutions.

Examples

Matrix Completion and Recommender Systems



Characteristics. Reconstruction of matrix from 500 observations. True rank is 30.

FrankWolfe.jl

a high-performance Julia package for Conditional Gradients

joint work with Mathieu Besançon and Alejandro Carderera

[Besançon et al., 2021]

Overview and Features

FrankWolfe.jl

In a nutshell.

- Implemented in Julia
- Open source under MIT License
- Generic numeric types: reduced (16, 32, 64 bits) and extended (128, GNU MP) precision, rationals
- Memory-saving mode, in-place gradient computations
- Scales well (solved some problems with 1B variables)
- Switch components - bring your own LMO / ∇f / step size.

Give it a try.

```
using Pkg
Pkg.add("FrankWolfe")
```

Example

FrankWolfe.jl

```
using LinearAlgebra
using FrankWolfe

n = 1000
xp = rand(n)

f(x) = norm(x - xp)^2

function grad!(storage, x)
    @. storage = 2 * (x - xp)
    return nothing
end


# create a L_1-norm ball of radius 2.5
lmo_radius = 2.5
lmo = FrankWolfe.LpNormLMO{Float64,1}(lmo_radius)

x0 = FrankWolfe.compute_extreme_point(lmo, zeros(n))

x_sol, _ = frank_wolfe(f, grad!, lmo, x0)
```

Thank you!

References I

F. Bach. On the effectiveness of Richardson extrapolation in machine learning. *arXiv preprint 2002.02835v3*, July 2020.

M. Besançon, A. Carderera, and S. Pokutta. FrankWolfe.jl: a high-performance and flexible toolbox for Frank-Wolfe algorithms and Conditional Gradients. *preprint*, 4 2021. URL https://arxiv.org/abs/2104.06675.

G. Braun and S. Pokutta. The matching polytope does not admit fully-polynomial size relaxation schemes. *Proceeedings of SODA*, 2015a. URL http://arxiv.org/abs/1403.6710.

G. Braun and S. Pokutta. The matching polytope does not admit fully-polynomial size relaxation schemes. *IEEE Transactions on Information Theory*, 61(10): 1–11, 2015b. URL http://arxiv.org/abs/1403.6710.

G. Braun, S. Pokutta, and D. Zink. Inapproximability of combinatorial problems via small LPs and SDPs. *Proceeedings of STOC*, 2015. URL http://arxiv.org/abs/1410.8816.

G. Braun, C. Guzmán, and S. Pokutta. Unifying Lower Bounds on the Oracle Complexity of Nonsmooth Convex Optimization. *IEEE Transactions of Information Theory*, 63(7):4709–4724, 2017a. URL http://arxiv.org/abs/1407.5144.

G. Braun, R. Jain, T. Lee, and S. Pokutta. Information-theoretic approximations of the nonnegative rank. *Computational Complexity*, 26(1):147–197, 2017b. URL http://eccc.hpi-web.de/report/2013/158.

G. Braun, S. Pokutta, and D. Zink. Lazifying Conditional Gradient Algorithms. *Proceedings of the International Conference on Machine Learning (ICML)*, 2017c. URL http://proceedings.mlr.press/v70/braun17a.

G. Braun, S. Pokutta, D. Tu, and S. Wright. Blended Conditional Gradients: the unconditioning of conditional gradients. *Proceedings of ICML*, 2019a. URL http://proceedings.mlr.press/v97/braun19a/braun19a.pdf.

G. Braun, S. Pokutta, and D. Zink. Lazifying Conditional Gradient Algorithms. *Journal of Machine Learning Research (JMLR)*, 20(71):1–42, 2019b. URL http://jmlr.org/papers/v20/18-114.html.

A. Carderera, J. Diakonikolas, C. Y. Lin, and S. Pokutta. Parameter-free Locally Accelerated Conditional Gradients. *to appear in Proceedings of ICML*, 2 2021.

C. W. Combettes and S. Pokutta. Complexity of Linear Minimization and Projection on Some Sets. *to appear in Operations Research Letters*, 1 2021+. URL https://arxiv.org/abs/2101.10040.

C. W. Combettes, C. Spiegel, and S. Pokutta. Projection-Free Adaptive Gradients for Large-Scale Optimization. *preprint*, 10 2020. URL https://arxiv.org/abs/2009.14114.

J. Diakonikolas, A. Carderera, and S. Pokutta. Locally Accelerated Conditional Gradients. *Proceedings of AISTATS*, 2020. URL https://arxiv.org/abs/1906.07867.

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

R. M. Freund, P. Grigas, and R. 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.

References II

D. Garber and E. Hazan. Playing non-linear games with linear oracles. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 420–428. IEEE, 2013.

D. Garber and O. Meshi. Linear-memory and decomposition-invariant linearly convergent conditional gradient algorithm for structured polytopes. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1001–1009. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6115-linear-memory-and-decomposition-invariant-linearly-convergent-conditional-gradient-algorithm-for-structured-pdf.

E. Hazan and S. Kale. Projection-free online learning. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

E. Hazan and H. Luo. Variance-reduced and projection-free stochastic optimization. In *International Conference on Machine Learning*, pages 1263–1271, 2016.

M. Jaggi. Revisiting Frank-Wolfe: projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*, pages 427–435, 2013.

T. Kerdreux, A. d'Aspremont, and S. Pokutta. Restarting Frank-Wolfe. *Proceedings of AISTATS*, 2019. URL http://proceedings.mlr.press/v89/kerdreux19a/kerdreux19a.pdf.

T. Kerdreux, A. d'Aspremont, and S. Pokutta. Projection-Free Optimization on Uniformly Convex Sets. *to appear in Proceedings of AISTATS*, 1 2021a. URL https://arxiv.org/abs/2004.11053.

T. Kerdreux, A. d'Aspremont, and S. Pokutta. Local and Global Uniform Convexity Conditions. *preprint*, 2 2021b. URL https://arxiv.org/abs/2102.05134.

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

S. Lacoste-Julien and M. 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 28*, pages 496–504. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5925-on-the-global-linear-convergence-of-frank-wolfe-optimization-variants.pdf.

G. Lan and Y. Zhou. Conditional gradient sliding for convex optimization. *SIAM Journal on Optimization*, 26(2):1379–1409, 2016.

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

J. Macdonald, S. Wäldchen, S. Hauch, and G. Kutyniok. A rate-distortion framework for explaining neural network decisions. *arXiv preprint arXiv:1905.11092*, 2019.

S. Pokutta, C. Spiegel, and M. Zimmer. Deep Neural Network Training with Frank-Wolfe. *preprint*, 10 2020. URL https://arxiv.org/abs/2010.07243.

S. J. Reddi, S. Sra, B. Póczos, and A. 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.

T. Rothvoss. The matching polytope has exponential extension complexity. In *Symposium on Theory of Computing*, pages 263–272, 2014.