Stochastic Frank-Wolfe for Constrained Finite-Sum Minimization

[1]**Geoffrey Négiar,** [2]**Gideon Dresdner,** [1]**Alicia Yi-Ting Tsai,** [1,5]**Laurent El Ghaoui,** [2]**Francesco Locatello,** [3]**Robert Freund,** [4]**Fabian Pedregosa**

June 12th, 2020. ICML, Online

[1]University of California, Berkeley [2]ETH, Zurich [3]MIT
[4]Google Research, Montréal [5]SumUp Analytics

Outline

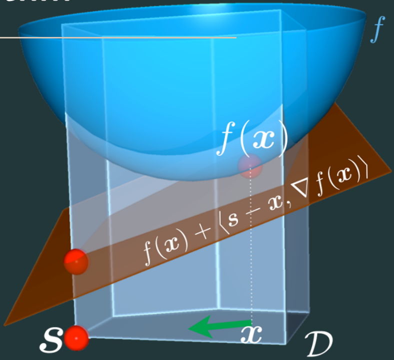
Motivation: Obtain a practical, fast version of Stochastic Frank-Wolfe.

Outline

Motivation: Obtain a practical, fast version of Stochastic Frank-Wolfe.

1. **Frank-Wolfe algorithm**. What is it and when is it used?
2. **Stochastic Frank-Wolfe**. Making Stochastic Frank-Wolfe practical: a primal-dual view.
3. **Results**. Convergence rates in theory and in practice.

The Frank-Wolfe algorithm



Frank-Wolfe: What is it?

Problem: smooth f, compact and convex $\mathcal{D}$

$$\arg\min_{\boldsymbol{x} \in \mathcal{D}} f(\boldsymbol{x})$$

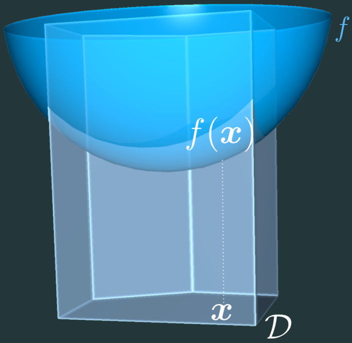
Algorithm 1: Frank-Wolfe (FW)

for $t = 0, 1 \ldots$ **do**

$\quad \boldsymbol{s}_t \in \arg\min_{\boldsymbol{s} \in \mathcal{D}} \langle \nabla f(\boldsymbol{x}_t), \boldsymbol{s} \rangle$

$\quad$ Find step-size γ_t.

$\quad \boldsymbol{x}_{t+1} = (1 - \gamma_t)\boldsymbol{x}_t + \gamma_t \boldsymbol{s}_t$



Frank-Wolfe: What is it?

Problem: smooth f, compact and convex $\mathcal{D}$

$$\arg\min_{\boldsymbol{x}\in\mathcal{D}} f(\boldsymbol{x})$$

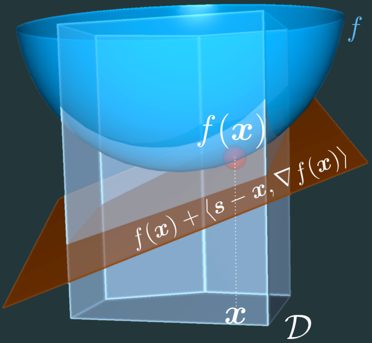
Algorithm 1: Frank-Wolfe (FW)

1 **for** $t = 0, 1 \ldots$ **do**

2 $\quad \boldsymbol{s}_t \in \arg\min_{\boldsymbol{s}\in\mathcal{D}} \langle \nabla f(\boldsymbol{x}_t), \boldsymbol{s}\rangle$

3 $\quad$ Find step-size γ_t.

4 $\quad \boldsymbol{x}_{t+1} = (1-\gamma_t)\boldsymbol{x}_t + \gamma_t \boldsymbol{s}_t$



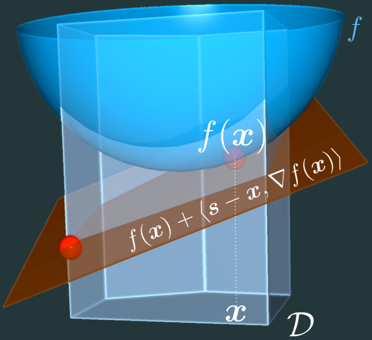
Frank-Wolfe: What is it?

Problem: smooth f, compact and convex $\mathcal{D}$

$$\operatorname*{arg\,min}_{\boldsymbol{x}\in\mathcal{D}} f(\boldsymbol{x})$$

Algorithm 1: Frank-Wolfe (FW)

for $t = 0, 1 \ldots$ **do**
- $\boldsymbol{s}_t \in \operatorname{arg\,min}_{\boldsymbol{s}\in\mathcal{D}} \langle \nabla f(\boldsymbol{x}_t), \boldsymbol{s}\rangle$
- Find step-size γ_t.
- $\boldsymbol{x}_{t+1} = (1-\gamma_t)\boldsymbol{x}_t + \gamma_t \boldsymbol{s}_t$



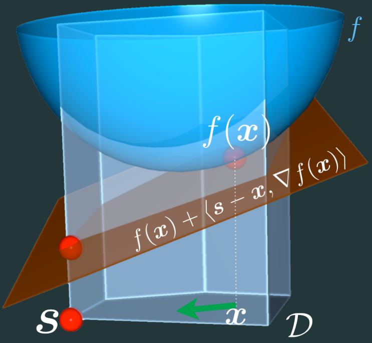
Frank-Wolfe: What is it?

Problem: smooth f, compact and convex $\mathcal{D}$

$$\operatorname*{arg\,min}_{\boldsymbol{x}\in\mathcal{D}} f(\boldsymbol{x})$$

Algorithm 1: Frank-Wolfe (FW)

for $t = 0, 1 \ldots$ **do**
    $\boldsymbol{s}_t \in \operatorname{arg\,min}_{\boldsymbol{s}\in\mathcal{D}} \langle \nabla f(\boldsymbol{x}_t), \boldsymbol{s}\rangle$
    Find step-size γ_t.
    $\boldsymbol{x}_{t+1} = (1-\gamma_t)\boldsymbol{x}_t + \gamma_t \boldsymbol{s}_t$



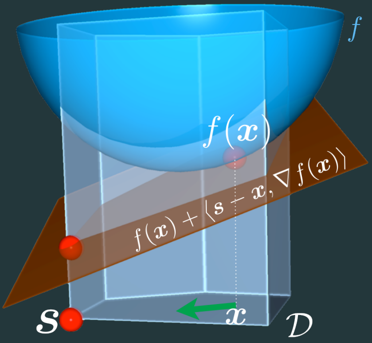
Frank-Wolfe: What is it?

Problem: smooth f, compact and convex $\mathcal{D}$

$$\arg\min_{\boldsymbol{x} \in \mathcal{D}} f(\boldsymbol{x})$$

Algorithm 1: Frank-Wolfe (FW)

for $t = 0, 1 \ldots$ **do**
- $\boldsymbol{s}_t \in \arg\min_{\boldsymbol{s} \in \mathcal{D}} \langle \nabla f(\boldsymbol{x}_t), \boldsymbol{s} \rangle$
- Find step-size γ_t.
- $\boldsymbol{x}_{t+1} = (1 - \gamma_t)\boldsymbol{x}_t + \gamma_t \boldsymbol{s}_t$



Frank-Wolfe: When do we use it?

- **Projection-free**. Linear subproblems vs. quadratic for projected gradient descent (PGD).

$$\min_{x \in \mathcal{D}} g^\top x \qquad \text{vs.} \qquad \min_{x \in \mathcal{D}} \|y - x\|_2^2$$

Frank-Wolfe: When do we use it?

- **Projection-free**. Linear subproblems vs. quadratic for projected gradient descent (PGD).

$$\min_{x \in \mathcal{D}} g^\top x \qquad \text{vs.} \qquad \min_{x \in \mathcal{D}} \|y - x\|_2^2$$

- Solution of linear subproblem: **extremal element** of $\mathcal{D}$.

Frank-Wolfe: When do we use it?

- **Projection-free**. Linear subproblems vs. quadratic for projected gradient descent (PGD).

$$\min_{x\in\mathcal{D}} g^\top x \qquad \text{vs.} \qquad \min_{x\in\mathcal{D}} \|y - x\|_2^2$$

- Solution of linear subproblem: **extremal element** of $\mathcal{D}$.
- **Sparse** representation: $\boldsymbol{x}_t$ convex combination of at most t elements.

Frank-Wolfe: When do we use it?

- **Projection-free**. Linear subproblems vs. quadratic for projected gradient descent (PGD).

$$\min_{x \in \mathcal{D}} g^\top x \qquad \text{vs.} \qquad \min_{x \in \mathcal{D}} \|y - x\|_2^2$$

- Solution of linear subproblem: **extremal element** of $\mathcal{D}$.
- **Sparse** representation: $\boldsymbol{x}_t$ convex combination of at most t elements.

Recent Applications

- Learning the structure of a neural network. Ping, Liu, and Ihler, 2016
- Attention mechanisms that enforce sparsity. Niculae, 2018
- ℓ_1-constrained problems with extreme number of features. Kerdreux, Pedregosa, and d'Aspremont, 2018

A practical issue for FW

- For large n (number of samples), we need a Stochastic variant of FW
- Naïve SGD-like algorithm fails in practice and in theory
- State of the art bounds on suboptimality after t iterations: $O(n/t)$ and $O(1/\sqrt[3]{t})$
  Lu and Freund, 2020; Mokhtari, Hassani, and Karbasi, 2018

A practical issue for FW

- For large n (number of samples), we need a Stochastic variant of FW
- Naïve SGD-like algorithm fails in practice and in theory
- State of the art bounds on suboptimality after t iterations: $O(n/t)$ and $O(1/\sqrt[3]{t})$
  Lu and Freund, 2020; Mokhtari, Hassani, and Karbasi, 2018

Can we do better?

Practical Stochastic Frank-Wolfe: a primal-dual point of view

Problem setting:

Let us add some structure: finite sum, and linear prediction

$$\textbf{OPT} : \min_{w \in \mathcal{C}} \ \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{x}_i^\top \boldsymbol{w})$$

- $f_i(\cdot)$ is the univariate loss function of observation/sample i for $i \in [n]$
- n is the number of observations/samples
- $\mathcal{C} \subset \mathbb{R}^d$ is a compact convex set
- d is the order (dimension) of the model variable $\boldsymbol{w}$

The particular structural dependence of the losses on $\boldsymbol{x}_i^\top \boldsymbol{w}$ is a model with "generalized linear structure" or "linear prediction"

Deterministic FW: Gradient Computation for OPT

OPT

$$f^* := \min_{\boldsymbol{w} \in \mathcal{C}} \; F(\boldsymbol{w}) = \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{x}_i^\top \boldsymbol{w})$$

Assumptions

- $f_i(\cdot)$ is L-smooth for $i \in [n]$: $\forall z, z'$, $|f_i'(z) - f_i'(z')| \leq L|z - z'|$
- Linear Minimization Oracle LMO($\boldsymbol{r}$): $\boldsymbol{s} \leftarrow \arg\min_{\boldsymbol{w} \in \mathcal{C}} \; \langle \boldsymbol{r}, \boldsymbol{w} \rangle$

Denote $\boldsymbol{X} := [\boldsymbol{x}_1^\top; \boldsymbol{x}_2^\top; \ldots; \boldsymbol{x}_n^\top]$

Gradient Computation

$\nabla F(\boldsymbol{w}) = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i \cdot f_i'(\boldsymbol{x}_i^\top \boldsymbol{w}) = \boldsymbol{X}^\top \boldsymbol{\alpha}$ where $\boldsymbol{\alpha}^i \leftarrow \frac{1}{n} f_i'(\boldsymbol{x}_i^\top \boldsymbol{w})$, $i \in [n]$

Gradient computation is $O(nd)$ operations (expensive when $n \gg 0 \ldots$)

Frank-Wolfe for OPT:

OPT

$$f^* := \min_{\boldsymbol{w} \in \mathcal{C}} \ F(\boldsymbol{w}) := \tfrac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{x}_i^\top \boldsymbol{w})$$

Frank-Wolfe algorithm for OPT:

Initialize at $\boldsymbol{w}_0 \in \mathcal{C}$, $t \leftarrow 0$.

At iteration t :

1. Compute $\nabla F(\boldsymbol{w}_{t-1})$:
   - $\alpha_t^i \leftarrow \frac{1}{n} f_i'(\boldsymbol{x}_i^\top \boldsymbol{w}_{t-1})$ for EVERY $i \in [n]$
   - $\boldsymbol{r}_t = \boldsymbol{X}^\top \alpha_t \ (= \nabla F(\boldsymbol{w}_{t-1}))$
2. Compute $\boldsymbol{s}_t \leftarrow \mathrm{LMO}(\boldsymbol{r}_t)$.
3. Set $\boldsymbol{w}_t \leftarrow \boldsymbol{w}_{t-1} + \gamma_t(\boldsymbol{s}_t - \boldsymbol{w}_{t-1})$, where $\gamma_t \in [0, 1]$.

Iteration cost is $O(nd)$ operations (expensive when $n \gg 0 \ldots$)

A Naïve Frank-Wolfe (SFW) Strategy

OPT

$$f^* := \min_{\boldsymbol{w} \in \mathcal{C}} F(\boldsymbol{w}) := \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{x}_i^\top \boldsymbol{w})$$

Frank-Wolfe algorithm for OPT:

Initialize at $\boldsymbol{w}_0 \in \mathcal{C}$, $t \leftarrow 0$.

At iteration t :

1. Compute $\nabla F(\boldsymbol{w}_{t-1})$:
   - $\alpha_t^i \leftarrow \frac{1}{n} f_i'(\boldsymbol{x}_i^\top \boldsymbol{w}_{t-1})$ for ONE $i \in [n]$ ($\alpha_t^j = 0$ for $j \neq i$)
   - $\boldsymbol{r}_t = \boldsymbol{X}^\top \alpha_t \; (= \boldsymbol{x}_i f_i'(\boldsymbol{x}_i^\top \boldsymbol{w}_{t-1}))$
2. Compute $\boldsymbol{s}_t \leftarrow \text{LMO}(\boldsymbol{r}_t)$.
3. Set $\boldsymbol{w}_t \leftarrow \boldsymbol{w}_{t-1} + \gamma_t(\boldsymbol{s}_t - \boldsymbol{w}_{t-1})$, where $\gamma_t \in [0, 1]$.

This approach does not work without growing the batch size [Hazan]

Our Frank-Wolfe (SFW) Strategy

OPT

$$f^* := \min_{\boldsymbol{w} \in \mathcal{C}} F(\boldsymbol{w}) := \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{x}_i^\top \boldsymbol{w})$$

Frank-Wolfe algorithm for OPT:

Initialize at $\boldsymbol{w}_0 \in \mathcal{C}$, $t \leftarrow 0$.

At iteration t :

1. Compute $\nabla F(\boldsymbol{w}_{t-1})$:
   - $\alpha_t^i \leftarrow \frac{1}{n} f_i'(\boldsymbol{x}_i^\top \boldsymbol{w}_{t-1})$ for ONE $i \in [n]$ ($\alpha_t^j = \alpha_{t-1}^j$ for $j \neq i$)
   - $\boldsymbol{r}_t = \boldsymbol{X}^\top \alpha_t \; (= \boldsymbol{r}_{t-1} + \boldsymbol{x}_i(\alpha_t^i - \alpha_{t-1}^i))$
2. Compute $\boldsymbol{s}_t \leftarrow \mathrm{LMO}(\boldsymbol{r}_t)$.
3. Set $\boldsymbol{w}_t \leftarrow \boldsymbol{w}_{t-1} + \gamma_t(\boldsymbol{s}_t - \boldsymbol{w}_{t-1})$, where $\gamma_t \in [0, 1]$.

Iteration cost is $O(d)$ operations! Memory cost is $O(d + n)$

Motivation: a Primal-Dual Lens for Constructing FW

Recall the definition of the *conjugate* of a function f:

$$f^*(\boldsymbol{\alpha}) := \max_{\boldsymbol{x} \in \mathrm{dom} f(\cdot)} \{\boldsymbol{\alpha}^\top \boldsymbol{x} - f(\boldsymbol{x})\}$$

- If f is a closed convex function, then $f^{**} = f$
- $f(\boldsymbol{x}) := \max_{\boldsymbol{\alpha} \in \mathrm{dom}\, f^*(\cdot)} \{\boldsymbol{\alpha}^\top \boldsymbol{x} - f^*(\boldsymbol{\alpha})\}$, and
- When f is differentiable, it holds that

$$\nabla f(\boldsymbol{x}) \leftarrow \boldsymbol{\alpha} \quad \text{where} \quad \boldsymbol{\alpha} \leftarrow \operatorname*{\mathbf{arg\,max}}_{\boldsymbol{\beta} \in \mathrm{dom} f^*(\cdot)} \{\boldsymbol{\beta}^\top \boldsymbol{x} - f^*(\boldsymbol{\beta})\} \ .$$

Motivation: a Primal-Dual Lens for Constructing FW

Using conjugacy we can reformulate **OPT** as:

$$\textbf{OPT}: \min_{\boldsymbol{w}\in\mathcal{C}} f(\boldsymbol{X}\boldsymbol{w}) = \min_{\boldsymbol{w}\in\mathcal{C}} \max_{\boldsymbol{\alpha}\in\mathbb{R}^n} \left\{ \mathcal{L}(\boldsymbol{w}, \boldsymbol{\alpha}) \stackrel{\text{def}}{=} \langle X\boldsymbol{w}, \boldsymbol{\alpha}\rangle - f^*(\boldsymbol{\alpha}) \right\}$$

Given $\boldsymbol{w}_{t-1}$ we construct the gradient of $f(\boldsymbol{X}\boldsymbol{w})$ at $\boldsymbol{w}_{t-1}$ by maximizing over the dual variable $\boldsymbol{\alpha}$:

$$\begin{aligned} \boldsymbol{\alpha}_t &\in \operatorname*{\mathbf{arg\,max}}_{\boldsymbol{\alpha}\in\mathbb{R}^n} \{\mathcal{L}(\boldsymbol{w}_{t-1}, \boldsymbol{\alpha}) = \langle \boldsymbol{X}\boldsymbol{w}_{t-1}, \boldsymbol{\alpha}\rangle - f^*(\boldsymbol{\alpha})\} \\ &\iff \nabla f(\boldsymbol{X}\boldsymbol{w}_{t-1}) = \boldsymbol{X}^\top \boldsymbol{\alpha}_t \end{aligned}$$

Then the LMO step corresponds to fixing the dual variable and minimizing over the primal variable $\boldsymbol{w}$:

$$\begin{aligned} \boldsymbol{s}_t &\leftarrow \operatorname*{\mathbf{arg\,min}}_{\boldsymbol{w}\in\mathcal{C}} \left\{ \mathcal{L}(\boldsymbol{w}, \boldsymbol{\alpha}_t) = \langle \boldsymbol{w}, \boldsymbol{X}^\top \boldsymbol{\alpha}_t\rangle - f^*(\boldsymbol{\alpha}_t) \right\} \\ &\iff \boldsymbol{s}_t \leftarrow \mathrm{LMO}(\boldsymbol{X}^\top \boldsymbol{\alpha}_t) \end{aligned}$$

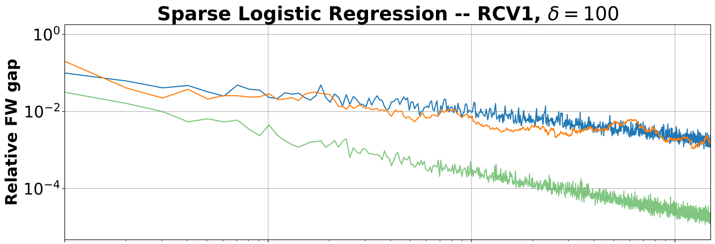
Results: Practice and Theory

Experiments RCV1

Problem: ℓ_1-constrained logistic regression

$$\operatorname*{arg\,min}_{\|\boldsymbol{x}\|_1 \le \alpha} \frac{1}{n} \sum_{i=1}^{n} \varphi(\boldsymbol{a}_i^\top \boldsymbol{x}, b_i) \quad \text{with } \varphi = \text{logistic loss.}$$

Dataset	dimension	#samples
RCV1	47236	20463

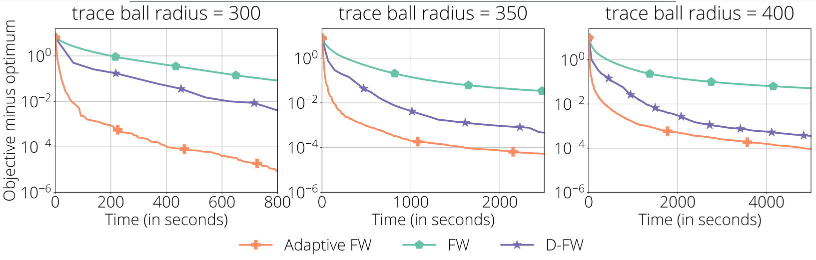


Experiments MovieLens 1M

Problem: trace-norm constrained robust matrix completion

$$\underset{\|\boldsymbol{x}\|_* \leq \alpha}{\arg\min} \frac{1}{|B|} \sum_{(i,j)\in B}^{n} h(\boldsymbol{X}_{i,j}, \boldsymbol{A}_{i,j}) \quad \text{with } h = \text{Huber loss.}$$

Dataset	dimension	density	$\overline{L}_t/L$
MovieLens 1M	22,393,987	0.04	1.1×10^{-2}



Theoretical guarantees: convex case

Define the ℓ_p norm "diameter" of $\mathcal{C}$ to be $D_p := \max_{\boldsymbol{w},\boldsymbol{v}\in\mathcal{C}} \|\boldsymbol{X}(\boldsymbol{w}-\boldsymbol{v})\|_p$

Theorem: Computational Complexity of Novel Stochastic Frank-Wolfe Algorithm

Let $H_0 \stackrel{\text{def}}{=} \|\boldsymbol{\alpha}_0 - \nabla f(\boldsymbol{X}\boldsymbol{w}_0)\|_1$ be the initial error of the gradient ∇f, and let the step-size rule be $\gamma_t = \frac{2}{t+2}$. For $t \geq 2$, it holds that:

$$
\begin{aligned}
\mathbb{E}[f(\boldsymbol{X}\boldsymbol{w}_t) - f^*] \quad \leq \quad & \frac{2(f(\boldsymbol{X}\boldsymbol{w}_0) - f^*)}{(t+1)(t+2)} \\
+ \quad & \left[2LD_2^2\left(\tfrac{1}{n}\right) + 8LD_1D_\infty\left(\tfrac{n-1}{n}\right)\right]\frac{t}{(t+1)(t+2)} \\
+ \quad & \frac{(2D_\infty H_0 + 64LD_1D_\infty)n^2}{(t+1)(t+2)} \,.
\end{aligned}
$$

Let us see what this bound is really about ...

Theoretical guarantees: convex case

ℓ_p norm "diameter" of $\mathcal{C}$ is $D_p := \max_{\boldsymbol{w},\boldsymbol{v}\in\mathcal{C}} \|\boldsymbol{X}(\boldsymbol{w}-\boldsymbol{v})\|_p$

Define Ratio $:= D_1/D_\infty$ and note that Ratio $\leq n$

The expected optimality gap bound is:

$$\frac{2(f(\boldsymbol{X}\boldsymbol{w}_0)-f^*)}{(t+1)(t+2)} + \left[2LD_2^2\left(\frac{1}{n}\right)+8LD_1D_\infty\left(\frac{n-1}{n}\right)\right]\left(\frac{1}{t}\right) + \frac{(2D_\infty H_0+64LD_1D_\infty)n^2}{(t+1)(t+2)}$$

$$= O\left(\frac{f(\boldsymbol{X}\boldsymbol{w}_0)-f^*}{t^2}\right) + O\left(\frac{LD_\infty^2(1+\text{Ratio})}{t}\right) + O\left((D_\infty H_0+LD_\infty^2\text{Ratio})\left(\frac{n^2}{t^2}\right)\right)$$

$$\leq O\left(\frac{LD_\infty^2\text{Ratio}}{t}\right) \leq O\left(\frac{n}{t}\right)$$

Theoretical guarantees: convex case

ℓ_p norm "diameter" of $\mathcal{C}$ is $D_p := \max_{\boldsymbol{w},\boldsymbol{v}\in\mathcal{C}} \|\boldsymbol{X}(\boldsymbol{w}-\boldsymbol{v})\|_p$

Define Ratio $:= D_1/D_\infty$ and note that Ratio $\leq n$

The expected optimality gap bound is:

$$\frac{2(f(\boldsymbol{X}\boldsymbol{w}_0)-f^*)}{(t+1)(t+2)} + \left[2LD_2^2\left(\tfrac{1}{n}\right) + 8LD_1D_\infty\left(\tfrac{n-1}{n}\right)\right]\left(\tfrac{1}{t}\right) + \frac{(2D_\infty H_0 + 64LD_1D_\infty)n^2}{(t+1)(t+2)}$$

$$= O\left(\frac{f(\boldsymbol{X}\boldsymbol{w}_0)-f^*}{t^2}\right) + O\left(\frac{LD_\infty^2(1+\text{Ratio})}{t}\right) + O\left(\left(D_\infty H_0 + LD_\infty^2\text{Ratio}\right)\left(\frac{n^2}{t^2}\right)\right)$$

$$\leq O\left(\frac{LD_\infty^2\text{Ratio}}{t}\right) \leq O\left(\frac{n}{t}\right)$$

Theoretical guarantees: convex case

ℓ_p norm "diameter" of $\mathcal{C}$ is $D_p := \max_{\boldsymbol{w},\boldsymbol{v}\in\mathcal{C}} \|\boldsymbol{X}(\boldsymbol{w}-\boldsymbol{v})\|_p$

Define Ratio $:= D_1/D_\infty$ and note that Ratio $\leq n$

The expected optimality gap bound is:

$$\frac{2(f(\boldsymbol{X}\boldsymbol{w}_0)-f^*)}{(t+1)(t+2)} + \left[2LD_2^2\left(\frac{1}{n}\right) + 8LD_1D_\infty\left(\frac{n-1}{n}\right)\right]\left(\frac{1}{t}\right) + \frac{(2D_\infty H_0 + 64LD_1D_\infty)n^2}{(t+1)(t+2)}$$

$$= O\left(\frac{f(\boldsymbol{X}\boldsymbol{w}_0)-f^*}{t^2}\right) + O\left(\frac{LD_\infty^2(1+\text{Ratio})}{t}\right) + O\left(\left(D_\infty H_0 + LD_\infty^2\text{Ratio}\right)\left(\frac{n^2}{t^2}\right)\right)$$

$$\leq O\left(\frac{LD_\infty^2\text{Ratio}}{t}\right) \leq O\left(\frac{n}{t}\right)$$

Theoretical guarantees: convex case

ℓ_p norm "diameter" of $\mathcal{C}$ is $D_p := \max_{\boldsymbol{w},\boldsymbol{v}\in\mathcal{C}} \|\boldsymbol{X}(\boldsymbol{w}-\boldsymbol{v})\|_p$

Define Ratio $:= D_1/D_\infty$ and note that Ratio $\leq n$

The expected optimality gap bound is:

$$\frac{2(f(\boldsymbol{X}\boldsymbol{w}_0)-f^*)}{(t+1)(t+2)} + \left[2LD_2^2\left(\tfrac{1}{n}\right) + 8LD_1D_\infty\left(\tfrac{n-1}{n}\right)\right]\left(\tfrac{1}{t}\right) + \frac{(2D_\infty H_0 + 64LD_1D_\infty)n^2}{(t+1)(t+2)}$$

$$= O\left(\frac{f(\boldsymbol{X}\boldsymbol{w}_0)-f^*}{t^2}\right) + O\left(\frac{LD_\infty^2(1+\text{Ratio})}{t}\right) + O\left(\left(D_\infty H_0 + LD_\infty^2\text{Ratio}\right)\left(\frac{n^2}{t^2}\right)\right)$$

$$\leq O\left(\frac{LD_\infty^2\text{Ratio}}{t}\right) \leq O\left(\frac{n}{t}\right)$$

Theoretical guarantees: convex case

ℓ_p norm "diameter" of $\mathcal{C}$ is $D_p := \max_{\boldsymbol{w},\boldsymbol{v}\in\mathcal{C}} \|\boldsymbol{X}(\boldsymbol{w}-\boldsymbol{v})\|_p$

Define Ratio $:= D_1/D_\infty$ and note that Ratio $\leq n$

The expected optimality gap bound is:

$$\frac{2(f(\boldsymbol{X}\boldsymbol{w}_0)-f^*)}{(t+1)(t+2)} + \left[2LD_2^2\left(\tfrac{1}{n}\right) + 8LD_1D_\infty\left(\tfrac{n-1}{n}\right)\right]\left(\tfrac{1}{t}\right) + \frac{(2D_\infty H_0 + 64LD_1D_\infty)n^2}{(t+1)(t+2)}$$

$$= O\left(\frac{f(\boldsymbol{X}\boldsymbol{w}_0)-f^*}{t^2}\right) + O\left(\frac{LD_\infty^2(1+\text{Ratio})}{t}\right) + O\left(\left(D_\infty H_0 + LD_\infty^2\text{Ratio}\right)\left(\frac{n^2}{t^2}\right)\right)$$

$$\leq O\left(\frac{LD_\infty^2\text{Ratio}}{t}\right) \leq O\left(\frac{n}{t}\right)$$

Theoretical guarantees: convex case

ℓ_p norm "diameter" of $\mathcal{C}$ is $D_p := \max_{\boldsymbol{w},\boldsymbol{v}\in\mathcal{C}} \|\boldsymbol{X}(\boldsymbol{w}-\boldsymbol{v})\|_p$

Define Ratio $:= D_1/D_\infty$ and note that Ratio $\leq n$

The expected optimality gap bound is:

$$\frac{2(f(\boldsymbol{X}\boldsymbol{w}_0) - f^*)}{(t+1)(t+2)} + \left[2LD_2^2\left(\tfrac{1}{n}\right) + 8LD_1D_\infty\left(\tfrac{n-1}{n}\right)\right]\left(\tfrac{1}{t}\right) + \frac{(2D_\infty H_0 + 64LD_1D_\infty)n^2}{(t+1)(t+2)}$$

$$= O\left(\frac{f(\boldsymbol{X}\boldsymbol{w}_0) - f^*}{t^2}\right) + O\left(\frac{LD_\infty^2(1+\text{Ratio})}{t}\right) + O\left(\left(D_\infty H_0 + LD_\infty^2\text{Ratio}\right)\left(\frac{n^2}{t^2}\right)\right)$$

$$\leq O\left(\frac{LD_\infty^2\text{Ratio}}{t}\right) \leq O\left(\frac{n}{t}\right)$$

Conclusion

- A practical, fast version of Stochastic Frank-Wolfe

Conclusion

- A practical, fast version of Stochastic Frank-Wolfe
- Hyperparameter-free

Conclusion

- A practical, fast version of Stochastic Frank-Wolfe
- Hyperparameter-free
- Implementation available in https://github.com/openopt/copt

Conclusion

- A practical, fast version of Stochastic Frank-Wolfe
- Hyperparameter-free
- Implementation available in
  `https://github.com/openopt/copt`
- Use FW when the structure of your problem demands it!

Thanks for your attention

References

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

Lu, Haihao and Robert Michael Freund (2020). "Generalized stochastic FrankWolfe algorithm with stochastic substitute gradient for structured convex optimization". In: *Math. Program.*

Mokhtari, Aryan, Hamed Hassani, and Amin Karbasi (2018). "Stochastic Conditional Gradient Methods: From Convex Minimization to Submodular Maximization". In: *ArXiv* abs/1804.09554.

Niculae, Vlad et al. (2018). "SparseMAP: Differentiable Sparse Structured Inference". In: *International Conference on Machine Learning*.

Ping, Wei, Qiang Liu, and Alexander T Ihler (2016). "Learning Infinite RBMs with Frank-Wolfe". In: *Advances in Neural Information Processing Systems*.