Optimization for Machine Learning
CS-439

Lecture 8: Frank-Wolfe algorithm, and Muon

Martin Jaggi

EPFL – `github.com/epfml/OptML_course`

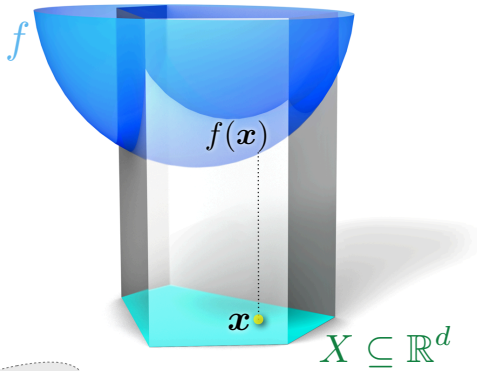
April 24, 2026

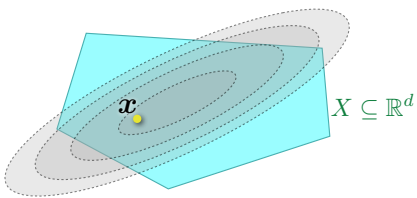
Frank-Wolfe

Constrained Optimization

Constrained Optimization Problem

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{x} \in \mathcal{X} \end{array}$$



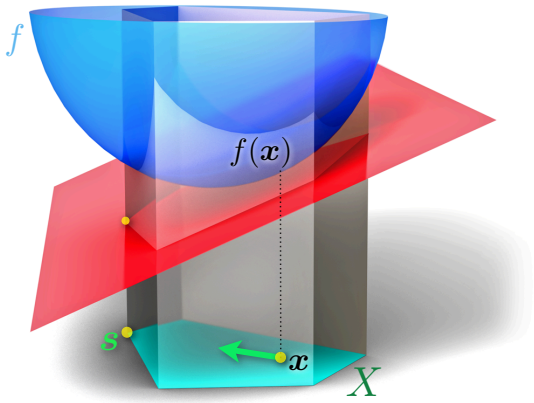


Frank-Wolfe Algorithm

Frank-Wolfe Algorithm:

$$\begin{aligned} \mathbf{s} &:= \mathrm{LMO}(\nabla f(\mathbf{x}_t)), \\ \mathbf{x}_{t+1} &:= (1-\gamma)\mathbf{x}_t + \gamma \mathbf{s}, \end{aligned}$$

for timesteps $t = 0, 1, \ldots,$ and stepsize $\gamma := \frac{2}{t+2}$.



Linear Minimization Oracle:

$$\mathrm{LMO}(\mathbf{g}) := \operatorname*{argmin}_{\mathbf{s}\in\mathcal{X}} \langle \mathbf{s}, \mathbf{g} \rangle$$

Properties

- Always feasible: $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t \in \mathcal{X}$.
  $\mathbf{x}_{t+1}$ is on line segment $[\mathbf{s}, \mathbf{x}_t]$, for $\gamma \in [0, 1]$.
- Reduces non-linear to linear optimization
- Projection-free
- Sparse iterates (in terms of corners $\mathbf{s}$ used)

Invented and analyzed 1956 by Marguerite Frank and Philip Wolfe.

Example

Lasso Regression

$$\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|^2 \quad s.t. \quad \|\mathbf{x}\|_1 \leq 1$$

L1-ball is the convex hull of the unit basis vectors:
$\mathcal{X} = \{\mathbf{x} \mid \|\mathbf{x}\|_1 \leq 1\} = \mathrm{conv}(\{\pm\mathbf{e}_1, \ldots, \pm\mathbf{e}_n\})$.

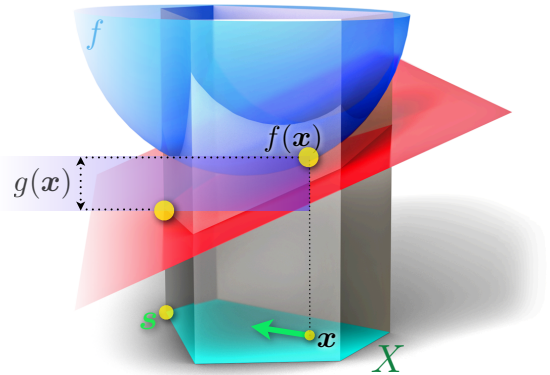
- $\nabla f(\mathbf{x}) = \mathbf{g} := A^\top(A\mathbf{x} - \mathbf{b})$
- $\mathrm{LMO}(\mathbf{g}) = -\mathrm{sign}(g_i)\mathbf{e}_i$ with $i := \operatorname*{argmax}_{i \in [n]} |g_i|$

simpler than projection onto L1-ball !

Duality Gap

Duality Gap

$$g(\mathbf{x}) := \langle \mathbf{x} - \mathbf{s}, \nabla f(\mathbf{x}) \rangle$$



Certificate for optimization quality:

$$\begin{aligned} g(\mathbf{x}) &= \max_{\mathbf{s} \in \mathcal{X}} \langle \mathbf{x} - \mathbf{s}, \nabla f(\mathbf{x}) \rangle \\ &\geq \langle \mathbf{x} - \mathbf{x}^\star, \nabla f(\mathbf{x}) \rangle \\ &\geq f(\mathbf{x}) - f(\mathbf{x}^\star) \end{aligned}$$

Stepsize variants

$$
\begin{aligned}
\gamma_t &:= \frac{2}{t+2}, \\
\gamma_t &:= \operatorname*{argmin}_{\gamma \in [0,1]} f\big((1-\gamma)\mathbf{x}_t + \gamma \mathbf{s}\big), && \text{(line-search)} \\
\gamma_t &:= \min\Big\{ \frac{g(\mathbf{x}_t)}{L \left\| \mathbf{s} - \mathbf{x}_t \right\|^2}, 1 \Big\}, && \text{(gap-based)}
\end{aligned}
$$

Convergence in $\mathcal{O}(1/\varepsilon)$ steps

Theorem

Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and smooth with parameter L, and $\mathbf{x}_0 \in \mathcal{X}$. Then choosing any of the above stepsizes, the Frank-Wolfe algorithm yields

$$f(\mathbf{x}_T) - f(\mathbf{x}^\star) \leq \frac{2L \operatorname{diam}(\mathcal{X})^2}{T+1}$$

Where $\operatorname{diam}(\mathcal{X}) := \max_{\mathbf{x},\mathbf{y} \in \mathcal{X}} \|\mathbf{x} - \mathbf{y}\|$ is the diameter of $\mathcal{X}$.

Proof of Convergence in $\mathcal{O}(1/\varepsilon)$ steps

Lemma

For a step $\mathbf{x}_{t+1} := \mathbf{x}_t + \gamma(\mathbf{s} - \mathbf{x}_t)$ *with arbitrary step-size* $\gamma \in [0, 1]$*, it holds that*

$$f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \gamma g(\mathbf{x}_t) + \tfrac{\gamma^2}{2} L \operatorname{diam}(\mathcal{X})^2 \ ,$$

if $\mathbf{s} = \operatorname{LMO}(\nabla f(\mathbf{x}_t))$*.*

Proof.

We write $\mathbf{x} := \mathbf{x}_t$, $\mathbf{y} := \mathbf{x}_{t+1} = \mathbf{x} + \gamma(\mathbf{s} - \mathbf{x})$. From the definition of smoothness of f, we have

$$\begin{aligned} f(\mathbf{y}) = \ & f(\mathbf{x} + \gamma(\mathbf{s} - \mathbf{x})) \\ \leq \ & f(\mathbf{x}) + \gamma \langle \mathbf{s} - \mathbf{x}, \nabla f(\mathbf{x}) \rangle + \tfrac{\gamma^2}{2} L \operatorname{diam}(\mathcal{X})^2 \ . \end{aligned}$$

The lemma follows by definition of the duality gap. □

Proof of Convergence in $\mathcal{O}(1/\varepsilon)$ steps

From the Lemma we know that for every step of FW, it holds that

$$f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \gamma g(\mathbf{x}_t) + \gamma^2 C,$$

if we chose $\gamma := \frac{2}{t+2}$ and write $C := \frac{1}{2} L \operatorname{diam}(\mathcal{X})^2$. This bound holds also for all mentioned line-search variants *(different LHS, same RHS)*.

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

$$\begin{aligned} h(\mathbf{x}_{t+1}) \leq & \quad h(\mathbf{x}_t) - \gamma g(\mathbf{x}_t) + \gamma^2 C \\ \leq & \quad h(\mathbf{x}_t) - \gamma h(\mathbf{x}_t) + \gamma^2 C \\ = & \quad (1-\gamma) h(\mathbf{x}_t) + \gamma^2 C \ , \end{aligned}$$

by the certificate property $h(\mathbf{x}) \leq g(\mathbf{x})$ of the duality gap.
The theorem then follows by induction (Lab 8, Exercise 1). □

Affine Invariance

Curvature Constant

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

Algorithm is invariant to scaling (affine transformations) of the input problem.

So is C_f.

(same as Newton, but here for constrained problems)

$$C_f \leq \frac{L}{2}\operatorname{diam}(\mathcal{X})^2 \qquad \text{for any norm!}$$

All proofs hold for C_f, instead of picking a particular $L\operatorname{diam}(\mathcal{X})^2$.

Convergence in Duality Gap

Theorem

Let $f : \mathbb{R}^d \to \mathbb{R}$ and $\mathcal{X}$ be convex with $C_{(f,\mathcal{X})} < \infty$, and $\mathbf{x}_0 \in \mathcal{X}$, $T \geq 2$. Then choosing any of the above stepsizes, the Frank-Wolfe algorithm yields a $t, 1 \leq t \leq T$ s.t.

$$g(\mathbf{x}_t) \leq \frac{27/2\, C_f}{T+1}$$

Proof.

Idea: not all gaps can be small (use Lemma again). □

Extensions and Use Cases

Extensions:

- Approximate LMO (of additive or multiplicative accuracy)
- LMO applied to stochastic gradient $\rightarrow$ stochastic FW (only works with momentum)
- Randomized LMO
- unconstrained problems (Matching Pursuit variants)

Use cases:

Whenever projection is more costly than solving a linear problem

- Lasso and other L1-constrained problems
- Matrix Completion: scalable algorithm
- Relaxation of combinatorial problems (e.g. matchings, network flows etc)
- Muon optimizer (highly performant for training deep neural networks)

Applications

recall: $\mathrm{LMO}(\mathbf{g}) := \operatorname*{argmin}_{\mathbf{s}\in\mathcal{X}} \langle \mathbf{s}, \mathbf{g} \rangle$

$$\mathcal{X} := conv(\mathcal{A})$$

Examples	$\mathcal{A}$	$\lvert\mathcal{A}\rvert$	d	LMO ($\mathbf{g}$)
L1-ball	$\{\pm\mathbf{e}_i\}$	$2d$	d	$\pm\mathbf{e}_i$ with $\operatorname{argmax}_i \lvert g_i\rvert$ (attains the $\lVert\mathbf{g}\rVert_\infty$ norm)
Simplex	$\{\mathbf{e}_i\}$	d	d	$\mathbf{e}_i$ with $\operatorname{argmin}_i g_i$
Norms	$\{\mathbf{x}, \lVert\mathbf{x}\rVert \leq 1\}$	∞	d	$\operatorname{argmin}_{\mathbf{s},\lVert\mathbf{s}\rVert\leq 1} \langle \mathbf{s}, \mathbf{g} \rangle$
Nuclear norm	$\{\mathbf{X}, \lVert\mathbf{X}\rVert_* \leq 1\}$	∞	d^2	(attains the Operator norm $\lVert\mathbf{g}\rVert_{\mathsf{op}}$)
Operator norm	$\{\mathbf{X}, \lVert\mathbf{X}\rVert_{\mathsf{op}} \leq 1\}$	∞	d^2	(attains the Nuclear norm $\lVert\mathbf{g}\rVert_*$)
Wavelets	..	∞	∞	..

Muon

Muon: MomentUm Orthogonalized by Newton-schulz

Algorithm. At each step t, with stochastic gradient $\mathbf{g}_t$:

$$\mathbf{M}_t := \mu\,\mathbf{M}_{t-1} + (1-\mu)\,\mathbf{g}_t \qquad \text{(momentum buffer)}$$
$$\mathbf{X}_{t+1} := \mathbf{X}_t - \eta\,\mathrm{orth}(\mathbf{M}_t) - \eta\lambda\mathbf{X}_t \qquad \text{(update, with optional weight decay } \lambda\text{)}$$

Orthogonalization. (Matrix-sign operation) If $\mathbf{M}_t = \mathbf{U}\Sigma\mathbf{V}^\top$ is the SVD, then

$$\mathrm{orth}(\mathbf{M}_t) \;=\; \mathbf{U}\mathbf{V}^\top.$$

All singular values are replaced by 1. Computed efficiently via Newton–Schulz iteration (5 steps, matmuls only — GPU-friendly, no SVD).

Scope in neural network training. Applied only to 2D weight matrices. Embeddings, biases&gains, and routers (in case of MoE) use AdamW. Stepsize $\eta \approx 0.02$. [JJB+24].

Why orthogonalize? Balancing per-layer update magnitudes

A layer's job: map input $\mathbf{a} \in \mathbb{R}^n$ to output activations $\mathbf{b} = \mathbf{X}\mathbf{a} \in \mathbb{R}^m$.

What matters for stability: How much an update $\Delta\mathbf{X}$ changes the output for a typical input:

$$\|\Delta\mathbf{b}\| = \|\Delta\mathbf{X} \cdot \mathbf{a}\| \;\leq\; \|\Delta\mathbf{X}\|_{\mathsf{op}} \cdot \|\mathbf{a}\|.$$

The operator norm (spectral norm) $\|\Delta\mathbf{X}\|_{\mathsf{op}} = \sigma_{\max}(\Delta\mathbf{X})$ is the right quantity — it controls the worst-case output change per unit input.

Problem with raw gradients. A (stochastic or deterministic) gradient $\mathbf{M}_t$ typically has singular values of very different magnitudes. A classical gradient step thus changes outputs of a layer differently than its input. Imbalance can amplify through depth.

Muon's key change. Replacing $\mathbf{M}_t$ with $\mathbf{U}\mathbf{V}^\top$ sets every singular value to 1: the update has spectral norm exactly 1 and acts uniformly across all directions. Each layer update contributes a controlled, bounded change to its output activations.

Muon as Frank–Wolfe

Frank–Wolfe over a convex set $\mathcal{X}$:

$$\mathbf{S}_t = \mathrm{LMO}(\mathbf{M}_t) := \arg\min_{\mathbf{S}\in\mathcal{X}} \langle \mathbf{S}, \mathbf{M}_t \rangle, \qquad \mathbf{X}_{t+1} = (1-\gamma)\mathbf{X}_t + \gamma\, \mathbf{S}_t.$$

Muon update (rearranged) is identical to FW with $\gamma = \eta\lambda$:

$$\mathbf{X}_{t+1} = \mathbf{X}_t - \eta\, \mathbf{U}\mathbf{V}^\top - \eta\lambda \mathbf{X}_t \;=\; (1-\eta\lambda)\mathbf{X}_t + \eta\lambda \left(-\tfrac{1}{\lambda}\mathbf{U}\mathbf{V}^\top\right).$$

Key identity. Orthogonalization is the LMO over the operator-norm unit ball. For $\mathbf{M}_t = \mathbf{U}\Sigma\mathbf{V}^\top$:

$$\boxed{-\mathbf{U}\mathbf{V}^\top \;=\; \arg\min_{\|\mathbf{S}\|_{\mathsf{op}}\le 1} \langle \mathbf{S}, \mathbf{M}_t \rangle}$$

The extreme points of $\{S : \|S\|_{\mathsf{op}} \le 1\}$ are the (semi-)orthogonal matrices.

Corollary: **Muon** with weight decay λ is identical to **Stochastic Frank–Wolfe** on

$$\min_{\mathbf{X}} F(\mathbf{X}) \quad \text{s.t.} \quad \|\mathbf{X}\|_{\mathsf{op}} \le 1/\lambda.$$

Weight decay sets constraint radius; orthogonalization solves the LMO [PXA+25, SW25]

Bibliography


Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein.
Muon: An optimizer for hidden layers in neural networks.
blogpost https://kellerjordan.github.io/posts/muon/, 2024.


Thomas Pethick, Wanyun Xie, Kimon Antonakopoulos, Zhenyu Zhu, Antonio Silveti-Falls, and Volkan Cevher.
Training deep learning models with norm-constrained lmos.
arXiv preprint arXiv:2502.07529, 2025.


Maria-Eleni Sfyraki and Jun-Kun Wang.
Lions and Muons: Optimization via Stochastic Frank-Wolfe.
arXiv preprint arXiv:2506.04192, 2025.