Frank-Wolfe / Conditional Gradient algorithm

Alexandre Gramfort

Master 2 Data Science, Univ. Paris Saclay
Optimisation for Data Science

Constrained optimization problem for FW

We consider the constrained optimization problem $(\mathcal{P})$:

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

- where f is a convex **objective function**
- $\mathcal{D}$ is the **domain** which we assume is a **convex** and **compact** set.

$\rightarrow$ Assuming f is smooth how would you solve this?
$\rightarrow$ Give me examples in machine learning of such a problem.

Constrained optimization problem for FW

We consider the constrained optimization problem ($\mathcal{P}$):

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

- where f is a convex **objective function**
- $\mathcal{D}$ is the **domain** which we assume is a **convex** and **compact** set.

→ Assuming f is smooth how would you solve this?
→ Give me examples in machine learning of such a problem.

Remark: Compactness of $\mathcal{D}$ is not necessary for projected gradient algo.
Remark: Frank-Wolfe algorithm is a projection free algorithm.
Remark: No assumption that $\mathcal{D}$ is of finite dimension.

Constrained optimization problem

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

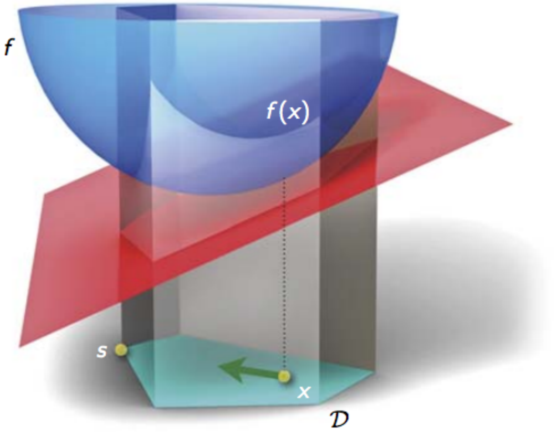


Image courtesy of Martin Jaggi (cf. [Jag13]).

Many applications

- network flows / transportation problems
- greedy selection and sparse optimization
- with wavelets (infinite-dimensional space)
- structured sparsity and structured prediction
- low-rank matrix factorizations, collaborative filtering
- total-variation-norm for image denoising
- submodular optimization
- boosting

Remark: Impressive revival in recent years in machine learning due to its low memory requirement and projection-free iterations

Application: Low-Rank Matrix Completion for collaborative filtering

Let $Y \in \mathbb{R}^{n \times m}$ be a partially observed data matrix.

Remark: Think of n as users and m as products and Y contains grades.

Ω denotes the entries of Y that are observed ($|\Omega| \ll n \times m$)

We want to solve:

$$\min_{X \in \mathbb{R}^{n \times m}} \sum_{(i,j) \in \Omega} (Y_{ij} - X_{ij})^2 \quad \text{s.t. } \|X\|_N \leq r .$$

where $\|X\|_N = \text{trace}\left(\sqrt{X^\top X}\right) = \sum_{i=1}^{\min\{m,n\}} \sigma_i(X)$.
It is the nuclear norm (sum of singular values).

Remark: $\|\cdot\|_N$ is a convex approximation of the rank.

Remark: $\mathcal{C} = \{X \in \mathbb{R}^{n \times m} \text{ s.t. } \|X\|_N \leq r\}$ convex.

LMO and linearization

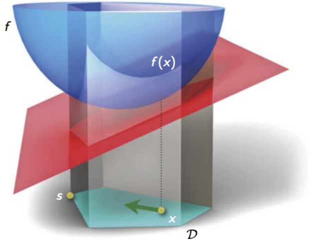
- Linearization of f at x:

$$f(s) \approx f(x) + \langle \nabla f(x), s - x \rangle = g_x(s)$$

- The Linear Minimization Oracle (LMO)

$$\mathrm{LMO}_{\mathcal{D}}(d) \triangleq \arg\min_{s \in \mathcal{D}} \langle d, s \rangle$$

$$\Rightarrow \mathrm{LMO}_{\mathcal{D}}(\nabla f(x)) = \arg\min_{s \in \mathcal{D}} g_x(s)$$



- **Idea:** For $\gamma \in [0, 1]$

$$x^{k+1} = \gamma \mathrm{LMO}_{\mathcal{D}}(\nabla f(x^k)) + (1 - \gamma) x_k$$

Remark: Step depends on domain $\mathcal{D}$ and $\nabla f(x^k)$, hence the name **conditional gradient**.

Frank-Wolfe / Conditional Gradient algorithm

$x^0 \in \mathcal{D}$
for $k = 0$ to n **do**
    $s = \mathrm{LMO}_{\mathcal{D}}(\nabla f(x^k))$
    $\gamma = \frac{2}{k+2}$
    $x^{k+1} = (1 - \gamma) x^k + \gamma s$
end for
return x^{n+1}

Frank-Wolfe / Conditional Gradient algorithm

$x^0 \in \mathcal{D}$
for $k = 0$ to n **do**
  $s = \mathrm{LMO}_{\mathcal{D}}(\nabla f(x^k))$
  $\gamma = \frac{2}{k+2}$
  $x^{k+1} = (1-\gamma)x^k + \gamma s$
end for
return x^{n+1}

With line search:

$$\gamma = \operatorname*{arg\,min}_{\gamma \in [0,1]} f((1-\gamma)x^k + \gamma s)$$

Convergence

- Marguerite Frank and Philip Wolfe showed in [FW56] that:

$$f(x^k) - f(x^*) \leq \mathcal{O}(1/k)$$

- Provided that:
  - f is smooth, convex and has some "curvature"
  - $\mathcal{D}$ is compact and convex

Remark: Same rates as projected gradient method but with simpler iterations. It is a projection free algorithm.

Remark: No free lunch: $\mathrm{LMO}_{\mathcal{D}}(\nabla f(x))$ needs to be easy.

Curvature constant vs. L-Liptschitz gradient

Let us define curvature constant C_f as:

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

Lemma

Let f be a convex and differentiable function with its gradient ∇f being Lipschitz-continuous w.r.t. some norm $\|\cdot\|$ over the domain $\mathcal{D}$ with Lipschitz-constant $L_{\|\cdot\|} > 0$. Then:

$$C_f \leq \mathrm{diam}_{\|\cdot\|}(\mathcal{D})^2 L_{\|\cdot\|} \ .$$

PROOF. Give it a try!

Remark: For L-smooth convex function on a compact convex domain: C_f exists

Convergence proof

Theorem

For f convex, with curvature C_f and $\mathcal{D}$ convex and compact. For each $k \geq 1$, the iterates x^k of the Frank-Wolfe algorithm satisfy

$$f(x^k) - f(x^*) \leq \frac{2C_f}{k+2} \ .$$

Convergence proof

PROOF. By definition of the C_f:

$$f(y) \leq f(x) + \gamma \underbrace{\langle s - x, \nabla f(x) \rangle}_{-g(x)} + \frac{\gamma^2}{2} C_f$$

for all $x, s \in \mathcal{D}$, $y = x + \gamma(s - x)$, $\gamma \in [0, 1]$.

Writing $h(x^k) = f(x^k) - f(x^*)$ for the error on objective, we have:

$$\begin{aligned} h(x^{k+1}) &\leq h(x^k) - \gamma g(x^k) + \frac{\gamma^2}{2} C_f && \text{(Definition of } C_f\text{)} \\ &\leq h(x^k) - \gamma h(x^k) + \frac{\gamma^2}{2} C_f && (h \leq g \text{ by convexity \& prop. of s}) \\ &= (1 - \gamma) h(x^k) + \frac{\gamma^2}{2} C_f . \end{aligned}$$

From here, the decrease rate follows from a simple lemma.

Convergence proof

Lemma

Suppose a sequence of numbers $(h_k)_k$ *satisfies*

$$h_{k+1} \leq (1 - \gamma^k) h_k + (\gamma^k)^2 C$$

for $\gamma^k = \frac{2}{k+2}$*, and* $k = 0, 1, \ldots,$ *and a constant* C*. Then*

$$h_k \leq \frac{4C}{k+2}, \; k = 0, 1, \ldots$$

PROOF. Trivial by induction.

Remark: [LJJ13] shows a linear/exponential convergence if f strongly convex and use line-search. It is like projected gradient descent but without projection!

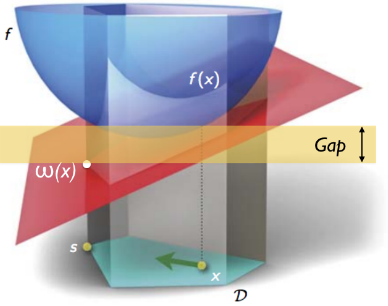
Optimality certificate (almost for free)

We solve:

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

Let:

$$\omega(x) = \min_{s \in \mathcal{D}} f(x) + \langle \nabla f(x), s - x \rangle$$



Lemma (Weak duality)

$\omega(x) \leq f(x^*) \leq f(x)$

So if $f(x) - \omega(x) \leq \epsilon$, x is an ϵ-solution.

Atomic Sets for fast LMO computation

If

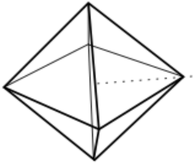
$$\mathcal{D} = \mathrm{conv}(\mathcal{A})$$

where $\mathcal{A}$ is a set (possibly infinite) of atoms/vectors. $\mathcal{A}$ is an "Atomic Set"

Then we have that $\forall x \in \mathcal{D}, \mathrm{LMO}_{\mathcal{D}}(\nabla \mathrm{f}(\mathrm{x})) \in \mathcal{A}$ (follows from the def. of a convex hull).

Example: ℓ_1 ball is an atomic set

$$\mathcal{D} = \mathrm{conv}(\{e_i | i \in [n]\} \cup \{-e_i | i \in [n]\})$$



So $\mathrm{LMO}_{\mathcal{D}}(\nabla f(x^k)) \in \{e_i | i \in [n]\} \cup \{-e_i | i \in [n]\}$.

Remark: We just need to find the smallest $\langle \nabla f(x_k), \pm e_i \rangle$

Let's practice

$\rightarrow$ frank_wolfe.ipynb notebook.

References

M. Frank and P. Wolfe.
An algorithm for quadratic programming.
Naval Res. Logis. Quart., 1956.

Martin Jaggi.
Revisiting frank-wolfe: Projection-free sparse convex optimization.
In *ICML*, volume 28, pages 427–435, June 2013.

S. Lacoste-Julien and M. Jaggi.
An affine invariant linear convergence analysis for frank-wolfe algorithms.
arXiv preprint arXiv:1312.7864, 2013.
https://arxiv.org/pdf/1312.7864.