The Frank-Wolfe Algorithm and its Applications

Aurélien Bellet
Télécom ParisTech

June 22, 2015
Outline

1. Introduction: Minimum Enclosing Ball

2. Frank-Wolfe
   ▶ Algorithm
   ▶ Convergence
   ▶ Comparison to gradient descent
   ▶ Variants

3. Applications
Minimum Enclosing Ball
Minimum Enclosing Ball
A classic problem in computational geometry

Given a set of $d$-dimensional points $S = \{x_1, \ldots, x_n\}$, find the smallest sphere $B(c_S, r_S)$ that contains $S$:

$$(c_S, r_S) = \arg \min_{c \in \mathbb{R}^d, r \geq 0} r \quad \text{s.t.} \quad \|c - x_i\| \leq r, \ i = 1, \ldots, n$$
Minimum Enclosing Ball
Approximation algorithm

Theorem (Coreset for MEB [Badoiu and Clarkson, 2008])

There is a subset \( S' \subseteq S \) of \( O(1/\epsilon) \) points such that an expansion by \( 1 + \epsilon \) of \( B(c_{S'}, r_{S'}) \) contains \( B(c_S, r_S) \). \( S' \) is called an \( \epsilon \)-coreset.

- Remarkable property: coreset size independent from \( d \) and \( n \)
- Found by greedy algorithm: initialize with random point, and iteratively add point furthest away from current center
- This is in fact a Frank-Wolfe algorithm applied to the dual problem of MEB
The Frank Wolfe algorithm (a.k.a. conditional gradient)
The Frank-Wolfe algorithm
Setup

- We are interested in the problem $\min_{\alpha \in \mathcal{D}} f(\alpha)$ where
  - $f$ convex and continuously differentiable
  - $\mathcal{D} \subset \mathbb{R}^d$ convex and compact

The curvature constant

Let $C_f$ be a constant such that

$$f(y) \leq f(\alpha) + \gamma \langle y - \alpha, \nabla f(\alpha) \rangle + \frac{\gamma^2}{2} C_f$$

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

- Measure of nonlinearity of $f$
- Bounded $C_f$ similar to $L$-Lipschitz gradient assumption:

$$C_f \leq \text{diam}\|\cdot\| (\mathcal{D})^2 L$$
The Frank-Wolfe algorithm

Algorithm

Let $\alpha^{(0)} \in \mathcal{D}$

for $k = 0, 1, \ldots$ do

$s^{(k)} = \arg \min_{s \in \mathcal{D}} \langle s, \nabla f(\alpha^{(k)}) \rangle$

$\alpha^{(k+1)} = (1 - \gamma)\alpha^{(k)} + \gamma s^{(k)}$

end for

- Consider the value $g(\alpha) := \max_{s \in \mathcal{D}} \langle \alpha - s, \nabla f(\alpha) \rangle$
  - Convexity of $f$ implies $g(\alpha) \geq f(\alpha) - f(\alpha^*)$
  - Special (and simplified) case of Fenchel duality

- This duality gap is given for free by the algorithm

Figure adapted from [Jaggi, 2013]
The Frank-Wolfe algorithm

Basic convergence

Convergence [Frank and Wolfe, 1956, Clarkson, 2010, Jaggi, 2013]

Let \( \gamma = 2/(k + 2) \). At any \( k \geq 1 \), \( \alpha^{(k)} \) is feasible and satisfies

\[
    f(\alpha^{(k)}) - f(\alpha^*) \leq \frac{2 C_f}{k + 2}
\]

Sketch of proof.

Writing \( h(\alpha^{(k)}) := f(\alpha^{(k)}) - f(\alpha^*) \), we have:

\[
    h(\alpha^{(k+1)}) \leq h(\alpha^{(k)}) - \gamma g(\alpha^{(k)}) + \frac{\gamma^2}{2} C_f
\]

\[
    \leq h(\alpha^{(k)}) - \gamma h(\alpha^{(k)}) + \frac{\gamma^2}{2} C_f
\]

\[
    = (1 - \gamma) h(\alpha^{(k)}) + \frac{\gamma^2}{2} C_f,
\]

and we obtain the decrease rate by induction.
The Frank-Wolfe algorithm
Comparison to gradient descent

- Recall Projected Gradient Descent (PGD):

  \[ \alpha^{(k+1)} = P_D \left( \alpha^{(k)} - \gamma \nabla f(\alpha^{(k)}) \right) \]

  - Requires a projection oracle \( P_D(\alpha) = \arg \min_{s \in \mathcal{D}} \| \alpha - s \| \)
  - Similar for proximal gradient

- Frank-Wolfe requires a linear minimization oracle:

  \[ \alpha = \arg \min_{s \in \mathcal{D}} \langle \alpha, s \rangle \]

  - Simpler in many cases (see examples later)

- Both have \( O(1/k) \) convergence (under current assumptions), but FW can be slower than GD in practice
The Frank-Wolfe algorithm
Improvements and variants

- Improved convergence
  - $O(1/k^2)$ when $f$ and $\mathcal{D}$ strongly convex
    [Garber and Hazan, 2015]
  - $O(\exp(-k))$ when $f$ is strongly convex and $x^* \in \text{int}(\mathcal{D})$
    [Guélat and Marcotte, 1986]
  - $O(\exp(-k))$ with away steps when $f$ is strongly convex
    [Lacoste-Julien and Jaggi, 2013]

- Many variants
  - Line-search, fully corrective [Jaggi, 2013]
  - Approximate linear subproblem, inexact gradient [Jaggi, 2013]
  - Away steps [Guélat and Marcotte, 1986]
  - Penalized version [Harchaoui et al., 2013]
  - Block-coordinate FW when $\mathcal{D} = \mathcal{D}_1 \times \cdots \times \mathcal{D}_m$
    [Lacoste-Julien et al., 2013]
Applications
Applications
Greedy optimization over atomic sets

Assume $\mathcal{D} = \text{conv}(\mathcal{A})$

- Solution of linear problem at a vertex of $\mathcal{D}$
- At each iteration, add an element $s \in \mathcal{A}$:

$$\alpha^{(k+1)} = (1 - \gamma)\alpha^{(k)} + \gamma s$$
Applications
Sparse approximation

- Optimization over $\ell_1$-norm ball: $D = \text{conv}(\{\pm e_i\}_{i=1}^n)$
- Linear problem: find maximum absolute entry of gradient
  - $O(n)$ complexity versus $O(n \log n)$ for projection
- Sparse iterates: $\alpha^{(0)} = 0 \implies \|\alpha^{(k)}\|_0 \leq k$
  - Accuracy-sparsity trade-off: $\epsilon$-approximation with $O(1/\epsilon)$ nonzero entries
  - Worst-case optimal [Jaggi, 2013]
- Similar for the simplex $\Delta_n = \text{conv}(\{e_i\}_{i=1}^n)$ [Clarkson, 2010]
Applications

Back to MEB

- Recall the MEB problem

\[(c_S, r_S) = \arg \min_{c \in \mathbb{R}^d, r \geq 0} r \quad \text{s.t. } \|c - x_i\| \leq r, i = 1, \ldots, n\]

- Denoting \(G = X^T X\), the Lagrangian dual is

\[
\min_{\alpha \in \Delta_n} \alpha^T G \alpha - \alpha^T \text{diag}(G)
\]

- The corresponding center is given by \(c = X\alpha\)

- \(i\)-th coordinate of the gradient: \(\|x_i - c\|^2 - c^T c\)
  - We recover the original algorithm (and guarantees)!

- Kernel people, any thoughts?
  - MEB can be kernelized
  - Can be used to get coreset results for SVM, and a \(O(n)\) training algorithm [Tsang et al., 2005, Tsang et al., 2007]
Applications
Back to MEB

- Recall the MEB problem

\[(c_S, r_S) = \arg \min_{c \in \mathbb{R}^d, r \geq 0} r \text{ s.t. } \|c - x_i\| \leq r, i = 1, \ldots, n\]

- Denoting \( G = X^T X \), the Lagrangian dual is

\[\min_{\alpha \in \Delta_n} \alpha^T G \alpha - \alpha^T \text{diag}(G)\]

- The corresponding center is given by \( c = X \alpha \)

- \( i \)-th coordinate of the gradient: \( \|x_i - c\|^2 - c^T c \)

- We recover the original algorithm (and guarantees)!

- Kernel people, any thoughts?

  - MEB can be kernelized
  - Can be used to get coreset results for SVM, and a \( O(n) \) training algorithm [Tsang et al., 2005, Tsang et al., 2007]
Applications

Low-rank approximation

- Optimization over the trace norm ball

\[ D = \text{conv} \left( \{uv^T : u \in \mathbb{R}^n, \|u\|_2 = 1, v \in \mathbb{R}^m, \|v\|_2 = 1 \} \right) \]

- Best convex relaxation of the rank
- Infinite number of vertices

- Linear problem: find largest singular vector of gradient
  - Trace norm is $\ell_1$ norm on the singular values
  - Square matrix: $O(n^2)$ complexity versus $O(n^3)$ for projection

- Low-rank iterates: \( M^{(0)} = 0 \implies \text{rank}(M^{(k)}) \leq k \)
  - Accuracy-rank trade-off: $\epsilon$-approximation of rank $O(1/\epsilon)$
  - Worst-case optimal [Jaggi, 2011]
Applications
My contributions

- Distributed sparse approximation [Bellet et al., 2015]
  - Find sparse combinations of $n$ distributed elements
    \[
    \min_{\alpha \in \mathbb{R}^n} f(\alpha) = g(A\alpha) \quad \text{s.t.} \quad \|\alpha\|_1 \leq \beta \quad (A \in \mathbb{R}^{d \times n})
    \]
  - Efficient distributed FW algorithm
  - Optimal accuracy-communication trade-off
  - Applications to LASSO, kernel SVM, Boosting

- Similarity learning for high-dimensional data [Liu et al., 2015]
  - Learn a huge but very sparse matrix $d \times d$
  - Specifically engineered domain with $O(d^2)$ atoms
  - Linear optimization is fast
  - Control of generalization-complexity trade-off by early stopping
  - Applications to similarity learning for textual data
Conclusion

- An alternative to gradient descent / proximal gradient
  - Efficient greedy algorithm for some domains
  - Provable accuracy-sparsity trade-off
  - Useful in large-scale learning problems

- Convergence may be slow when solution is not so sparse

- Some things still to be addressed
  - Nonsmooth setting?
  - Stochastic Frank-Wolfe?
  - ...
References I

Optimal core-sets for balls.

A Distributed Frank-Wolfe Algorithm for Communication-Efficient Sparse Learning.
In *SDM*.

Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm.

An algorithm for quadratic programming.

Faster Rates for the Frank-Wolfe Method over Strongly-Convex Sets.
In *ICML*.

Some comments on Wolfe’s away step.
References II


