

# Conditional Gradient (Frank-Wolfe) Method

Ryan Tibshirani  
Convex Optimization 10-725/36-725

## Last time: coordinate descent

For the problem

$$\min_x g(x) + \sum_{i=1}^n h_i(x_i)$$

with  $g$  convex and smooth and each  $h_i$  convex, can use **coordinate descent**, which begins with an initial points  $x^{(0)}$  and repeats:

$$x_1^{(k)} \in \operatorname{argmin}_{x_1} f(x_1, x_2^{(k-1)}, x_3^{(k-1)}, \dots, x_n^{(k-1)})$$

$$x_2^{(k)} \in \operatorname{argmin}_{x_2} f(x_1^{(k)}, x_2, x_3^{(k-1)}, \dots, x_n^{(k-1)})$$

...

$$x_n^{(k)} \in \operatorname{argmin}_{x_n} f(x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, \dots, x_n)$$

for  $k = 1, 2, 3, \dots$ . The above minimizations can also be replaced by proximal gradient steps

## Strengths:

- Relatively **simple** and can be surprisingly **efficient and scalable** when updates are implemented carefully
- When combined with a pathwise approach, and when utilizing active set tricks, takes advantage of **low-dimensional** structure inherent in a problem

## Weaknesses/unknowns:

- Not always applicable, when nonsmooth parts do not separate
- Not generically **parallelizable**, as updates are “one-at-a-time”
- Precise **rates** for cyclic coordinate descent **not well-understood** (especially for exact coordinatewise minimization)

# Conditional gradient method

Consider the constrained problem

$$\min_x f(x) \quad \text{subject to } x \in C$$

where  $f$  is convex and smooth, and  $C$  is convex. Recall **projected gradient descent** chooses an initial  $x^{(0)}$ , repeats for  $k = 1, 2, 3, \dots$

$$x^{(k)} = P_C(x^{(k-1)} - t_k \nabla f(x^{(k-1)}))$$

where  $P_C$  is the projection operator onto the set  $C$

This was a special case of proximal gradient descent, motivated by a local quadratic expansion of  $f$ :

$$x^{(k)} = P_C \left( \operatorname{argmin}_y \nabla f(x^{(k-1)})^T (y - x^{(k-1)}) + \frac{1}{2t} \|y - x^{(k-1)}\|_2^2 \right)$$

The **conditional gradient method**, also known as the Frank-Wolfe method, uses a local linear expansion of  $f$ :

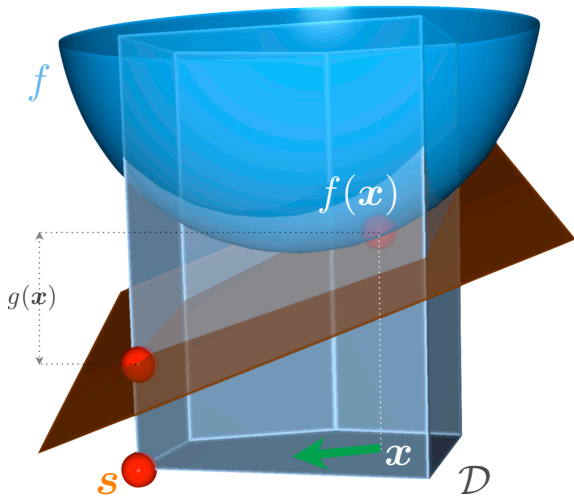
$$s^{(k-1)} \in \operatorname{argmin}_{s \in C} \nabla f(x^{(k-1)})^T s$$
$$x^{(k)} = (1 - \gamma_k)x^{(k-1)} + \gamma_k s^{(k-1)}$$

Note that there is no projection; update is solved directly over the constraint set  $C$

The default choice for step sizes is  $\gamma_k = 2/(k+1)$ ,  $k = 1, 2, 3, \dots$ . For any choice  $0 \leq \gamma_k \leq 1$ , we see that  $x^{(k)} \in C$  by convexity. Can also think of the update as

$$x^{(k)} = x^{(k-1)} + \gamma_k (s^{(k-1)} - x^{(k-1)})$$

i.e., we are moving less and less in the direction of the linearization minimizer as the algorithm proceeds



(From Jaggi 2011)

## Norm constraints

What happens when  $C = \{x : \|x\| \leq t\}$  for a norm  $\|\cdot\|$ ? Then

$$\begin{aligned} s &\in \operatorname{argmin}_{\|s\| \leq t} \nabla f(x^{(k-1)})^T s \\ &= -t \cdot \left( \operatorname{argmax}_{\|s\| \leq 1} \nabla f(x^{(k-1)})^T s \right) \\ &= -t \cdot \partial \|\nabla f(x^{(k-1)})\|_* \end{aligned}$$

where  $\|\cdot\|_*$  is the corresponding dual norm. In other words, if we know how to compute **subgradients of the dual norm**, then we can easily perform Frank-Wolfe steps

A key to Frank-Wolfe: this can often be simpler or cheaper than projection onto  $C = \{x : \|x\| \leq t\}$ . Also often simpler or cheaper than the prox operator for  $\|\cdot\|$

# Outline

Today:

- Examples
- Convergence analysis
- Properties and variants
- Path following



## Example: $\ell_1$ regularization

For the  $\ell_1$ -regularized problem

$$\min_x f(x) \quad \text{subject to} \quad \|x\|_1 \leq t$$

we have  $s^{(k-1)} \in -t\partial\|\nabla f(x^{(k-1)})\|_\infty$ . Frank-Wolfe update is thus

$$i_{k-1} \in \operatorname{argmax}_{i=1,\dots,p} |\nabla_i f(x^{(k-1)})|$$

$$x^{(k)} = (1 - \gamma_k)x^{(k-1)} - \gamma_k t \cdot \operatorname{sign}(\nabla_{i_{k-1}} f(x^{(k-1)})) \cdot e_{i_{k-1}}$$

Like greedy coordinate descent!

Note: this is a lot simpler than  $\ell_1$  projection onto the  $\ell_1$  ball, though both require  $O(n)$  operations

## Example: $\ell_p$ regularization

For the  $\ell_p$ -regularized problem

$$\min_x f(x) \quad \text{subject to} \quad \|x\|_p \leq t$$

for  $1 \leq p \leq \infty$ , we have  $s^{(k-1)} \in -t\partial\|\nabla f(x^{(k-1)})\|_q$ , where  $p, q$  are dual, i.e.,  $1/p + 1/q = 1$ . Claim: can choose

$$s_i^{(k-1)} = -\alpha \cdot \text{sign}(\nabla f_i(x^{(k-1)})) \cdot |\nabla f_i(x^{(k-1)})|^{p/q}, \quad i = 1, \dots, n$$

where  $\alpha$  is a constant such that  $\|s^{(k-1)}\|_q = t$  (check this!), and then Frank-Wolfe updates are as usual

Note: this is a lot simpler **projection onto the  $\ell_p$  ball**, for general  $p$ ! Aside from special cases ( $p = 1, 2, \infty$ ), these projections cannot be directly computed (must be treated as an optimization)

## Example: trace norm regularization

For the **trace-regularized** problem

$$\min_X f(X) \quad \text{subject to} \quad \|X\|_{\text{tr}} \leq t$$

we have  $S^{(k-1)} \in -t \|\nabla f(X^{(k-1)})\|_{\text{op}}$ . Claim: can choose

$$S^{(k-1)} = -t \cdot uv^T$$

where  $u, v$  are leading left, right singular vectors of  $\nabla f(X^{(k-1)})$  (check this!), and then Frank-Wolfe updates are as usual

Note: this is a lot simpler and more efficient than **projection onto the trace norm ball**, which requires a singular value decomposition!

## Constrained and Lagrange forms

Recall that solution of the **constrained** problem

$$\min_x f(x) \quad \text{subject to} \quad \|x\| \leq t$$

are equivalent to those of the **Lagrange** problem

$$\min_x f(x) + \lambda \|x\|$$

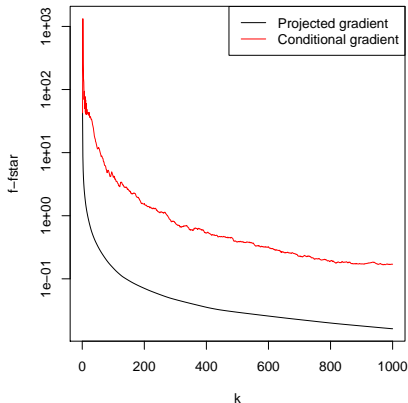
as we let the tuning parameters  $t$  and  $\lambda$  vary over  $[0, \infty]$ . Typically in statistics and ML problems, we would just solve **whichever form is easiest**, over wide range of parameter values

So we should also compare the Frank-Wolfe updates under  $\|\cdot\|$  to the proximal operator of  $\|\cdot\|$

- $\ell_1$  norm: Frank-Wolfe update scans for maximum of gradient; proximal operator soft-thresholds the gradient step; both use  $O(n)$  flops
- $\ell_p$  norm: Frank-Wolfe update computes raises each entry of gradient to power and sums, in  $O(n)$  flops; proximal operator not generally directly computable
- Trace norm: Frank-Wolfe update computes top left and right singular vectors of gradient; proximal operator soft-thresholds the gradient step, requiring a singular value decomposition

Many other regularizers yield efficient Frank-Wolfe updates, e.g., special polyhedra or cone constraints, sum-of-norms (group-based) regularization, atomic norms. See Jaggi (2011)

Comparing projected and conditional gradient for constrained lasso problem, with  $n = 100$ ,  $p = 500$ :



We will see that Frank-Wolfe methods match convergence rates of known first-order methods; but in practice they can be **slower to converge to high accuracy** (note: fixed step sizes here, line search would probably improve convergence)

## Duality gap

Frank-Wolfe iterations admit a very natural **duality gap** (truly, a suboptimality gap):

$$\max_{s \in C} \nabla f(x^{(k-1)})^T (x^{(k-1)} - s)$$

This is an upper bound on  $f(x^{(k-1)}) - f^\star$

Proof: by the first-order condition for convexity

$$f(s) \geq f(x^{(k-1)}) + \nabla f(x^{(k-1)})^T (y - x^{(k-1)})$$

Minimizing both sides over all  $s \in C$  yields

$$f^\star \geq f(x^{(k-1)}) + \min_{s \in C} \nabla f(x^{(k-1)})^T (y - x^{(k-1)})$$

Rearranged, this gives the duality gap above

Note that

$$\max_{s \in C} \nabla f(x^{(k-1)})^T (x^{(k-1)} - s) = \nabla f(x^{(k-1)})^T (x^{(k-1)} - s^{(k-1)})$$

so this quantity comes **directly from** the Frank-Wolfe update. Why do we call it “duality gap”? Rewrite original problem as

$$\min_x f(x) + I_C(x)$$

where  $I_C$  is the indicator function of  $C$ . The dual problem is

$$\max_u -f^*(u) - I_C^*(-u)$$

where  $I_C^*$  is the support function of  $C$ . Duality gap at  $x, u$  is

$$f(x) + f^*(u) + I_C^*(-u) \geq x^T u + I_C^*(-u)$$

Evaluated at  $x = x^{(k-1)}$ ,  $u = \nabla f(x^{(k-1)})$ , this gives claimed gap



## Convergence analysis

Following Jaggi (2011), define the **curvature constant** of  $f$  over  $C$ :

$$M = \max_{\substack{x, s, y \in C \\ y = (1-\gamma)x + \gamma s}} \frac{2}{\gamma^2} \left( f(y) - f(x) - \nabla f(x)^T (y - x) \right)$$

(Above we restrict  $\gamma \in [0, 1]$ .) Note that  $\kappa = 0$  when  $f$  is linear. The quantity  $f(y) - f(x) - \nabla f(x)^T (y - x)$  is called the **Bregman divergence** defined by  $f$

**Theorem:** Conditional gradient method using fixed step sizes  $\gamma_k = 2/(k+1)$ ,  $k = 1, 2, 3, \dots$  satisfies

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

Hence the number of iterations needed to achieve  $f(x^{(k)}) - f^* \leq \epsilon$  is  $O(1/\epsilon)$

This matches the known rate for projected gradient descent when  $\nabla f$  is Lipschitz, but how do the assumptions compare?. In fact, if  $\nabla f$  is Lipschitz with constant  $L$  then  $M \leq \text{diam}^2(C) \cdot L$ , where

$$\text{diam}(C) = \max_{x,s \in C} \|x - s\|_2$$

To see this, recall that  $\nabla f$  Lipschitz with constant  $L$  means

$$f(y) - f(x) - \nabla f(x)^T(y - x) \leq \frac{L}{2} \|y - x\|_2^2$$

Maximizing over all  $y = (1 - \gamma)x + \gamma s$ , and multiplying by  $2/\gamma^2$ ,

$$M \leq \max_{\substack{x,s,y \in C \\ y=(1-\gamma)x+\gamma s}} \frac{2}{\gamma^2} \cdot \frac{L}{2} \|y - x\|_2^2 = \max_{x,s \in C} L \|x - s\|_2^2$$

and the bound follows. Essentially, assuming a bounded curvature is **no stronger** than what we assumed for proximal gradient

## Basic inequality

The **key inequality** used to prove the Frank-Wolfe convergence rate is:

$$f(x^{(k)}) \leq f(x^{(k-1)}) - \gamma_k g(x^{(k-1)}) + \frac{\gamma_k^2}{2} M$$

Here  $g(x) = \max_{s \in C} \nabla f(x)^T (x - s)$  is the duality gap discussed earlier. The rate follows from this inequality, using induction

Proof: write  $x^+ = x^{(k)}$ ,  $x = x^{(k-1)}$ ,  $s = s^{(k-1)}$ ,  $\gamma = \gamma_k$ . Then

$$\begin{aligned} f(x^+) &= f(x + \gamma(s - x)) \\ &\leq f(x) + \gamma \nabla f(x)^T (s - x) + \frac{\gamma^2}{2} M \\ &= f(x) - \gamma g(x) + \frac{\gamma^2}{2} M \end{aligned}$$

Second line used definition of  $M$ , and third line the definition of  $g$

## Affine invariance

Important property of Frank-Wolfe: its updates are **affine invariant**.  
Given nonsingular  $A : \mathbb{R}^n \rightarrow \mathbb{R}^n$ , define  $x = Ax'$ ,  $h(x') = f(Ax')$ .  
Then Frank-Wolfe on  $h(x')$  proceeds as

$$s' = \operatorname{argmin}_{z \in A^{-1}C} \nabla h(x')^T z$$
$$(x')^+ = (1 - \gamma)x' + \gamma s'$$

Multiplying by  $A$  reveals precisely the same Frank-Wolfe update as would be performed on  $f(x)$

In fact, even the convergence analysis is affine invariant. Note that the curvature constant  $M$  of  $h$  is

$$M = \max_{\substack{x', s', y' \in A^{-1}C \\ y' = (1-\gamma)x' + \gamma s'}} \frac{2}{\gamma^2} \left( h(y') - h(x') - \nabla h(x')^T (y' - x') \right)$$

matching that of  $f$ , because  $\nabla h(x')^T (y' - x') = \nabla f(x)^T (y - x)$

## Inexact updates

Jaggi (2011) also analyzes **inexact Frank-Wolfe updates**. That is, suppose we choose  $s^{(k-1)}$  so that

$$\nabla f(x^{(k-1)})^T s^{(k-1)} \leq \min_{s \in C} \nabla f(x^{(k-1)})^T s + \frac{M\gamma_k}{2} \cdot \delta$$

where  $\delta \geq 0$  is our inaccuracy parameter. Then we basically attain the same rate

**Theorem:** Conditional gradient method using fixed step sizes  $\gamma_k = 2/(k+1)$ ,  $k = 1, 2, 3, \dots$ , and inaccuracy parameter  $\delta \geq 0$ , satisfies

$$f(x^{(k)}) - f^* \leq \frac{2M}{k+1}(1 + \delta)$$

Note: the optimization error at step  $k$  is  $M\gamma_k/2 \cdot \delta$ . Since  $\gamma_k \rightarrow 0$ , we require the errors to vanish

## Two variants

Two important variants of the conditional gradient method:

- **Line search:** instead of fixing  $\gamma_k = 2/(k+1)$ ,  $k = 1, 2, 3, \dots$ , use exact line search for the step sizes

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

at each  $k = 1, 2, 3, \dots$ . Or, we could use backtracking

- **Fully corrective:** directly update according to

$$x^{(k)} = \operatorname{argmin}_y f(y) \quad \text{subject to} \quad y \in \operatorname{conv}\{x^{(0)}, s^{(0)}, \dots, s^{(k-1)}\}$$

Can make much better progress, but is also quite a bit harder

Both variants have the same  $O(1/\epsilon)$  complexity, measured by the number of iterations

## Path following

Given the norm constrained problem

$$\min_x f(x) \quad \text{subject to} \quad \|x\| \leq t$$

the Frank-Wolfe algorithm can be used for **path following**, i.e., can produce an (approximate) solution path  $\hat{x}(t)$ ,  $t \geq 0$ . Beginning at  $t_0 = 0$  and  $x^*(0) = 0$ , we fix parameters  $\epsilon, m > 0$ , then repeat for  $k = 1, 2, 3, \dots$ :

- Calculate

$$t_k = t_{k-1} + \frac{(1 - 1/m)\epsilon}{\|\nabla f(\hat{x}(t_{k-1}))\|_*}$$

and set  $\hat{x}(t) = \hat{x}(t_{k-1})$  for all  $t \in (t_{k-1}, t_k)$

- Compute  $\hat{x}(t_k)$  by running Frank-Wolfe at  $t = t_k$ , terminating when the duality gap is  $\leq \epsilon/m$

This is a simplification of the strategy given in Giesen et al. (2012)

With this path following strategy, we are guaranteed that

$$f(\hat{x}(t)) - f(x^*(t)) \leq \epsilon \quad \text{for all } t \text{ visited}$$

i.e., we produce a (piecewise-constant) path with suboptimality gap **uniformly bounded** by  $\epsilon$ , over all  $t$

To see this, it helps to rewrite the Frank-Wolfe duality gap as

$$g_t(x) = \max_{\|s\| \leq 1} \nabla f(x)^T (x - s) = \nabla f(x)^T x + t \|\nabla f(x)\|_*$$

This is a linear function of  $t$ . Hence if  $g_t(x) \leq \epsilon/m$ , then we can increase  $t$  until  $t^+ = t + (1 - 1/m)\epsilon/\|\nabla f(x)\|_*$ , because at this value

$$g_{t^+}(x) = \nabla f(x)^T x + t \|\nabla f(x)\|_* + \epsilon - \epsilon/m \leq \epsilon$$

i.e., the duality gap remains  $\leq \epsilon$  for the same  $x$ , between  $t$  and  $t^+$



## References

- K. Clarkson (201), “Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm”
- J. Giesen and M. Jaggi and S. Laue, S. (2012), “Approximating parametrized convex optimization problems”
- M. Jaggi (2011), “Sparse convex optimization methods for machine learning”
- M. Jaggi (2011), “Revisiting Frank-Wolfe: projection-free sparse convex optimization”
- M. Frank and P. Wolfe (2011), “An algorithm for quadratic programming”
- R. J. Tibshirani (2015), “A general framework for fast stagewise algorithms”