Conditional Gradient (Frank-Wolfe) Method

Lecturer: Aarti Singh
Co-instructor: Pradeep Ravikumar

Convex Optimization 10-725/36-725

Outline

Today:

- Conditional gradient method
- Convergence analysis
- Properties and variants

So far ...

Unconstrained optimization

- Gradient descent
- Conjugate gradient method
- Accelerated gradient methods
- Newton and Quasi-newton methods
- Trust region methods
- Proximal gradient descent

So far ...

Unconstrained optimization

- Gradient descent
- Conjugate gradient method
- Accelerated gradient methods
- Newton and Quasi-newton methods
- Trust region methods
- Proximal gradient descent

Constrained optimization

- Projected gradient descent
- Conditional gradient (Frank-Wolfe) method - today
- ...

Projected gradient descent

Consider the constrained problem

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

where f is convex and smooth, and C is convex.

Recall projected gradient descent: choose an initial $x^{(0)}$, and for $k = 1, 2, 3, \ldots$

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

where P_C is the projection operator onto the set C

Projected gradient descent

Consider the constrained problem

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

where f is convex and smooth, and C is convex.

Recall projected gradient descent: choose an initial $x^{(0)}$, and for $k = 1, 2, 3, \ldots$

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

where P_C is the projection operator onto the set C

This was a special case of proximal gradient descent.

Gradient, proximal and projected gradient descent were motivated by a local quadratic expansion of f:

$$f(y) \approx f(x) + \nabla f(x)^T (y - x) + \frac{1}{2t}(y - x)^T (y - x)$$

leading to

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

Gradient, proximal and projected gradient descent were motivated by a local quadratic expansion of f:

$$f(y) \approx f(x) + \nabla f(x)^T (y - x) + \frac{1}{2t}(y - x)^T (y - x)$$

leading to

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

Newton method improved the quadratic expansion using Hessian of f (can do projected Newton too):

$$f(y) \approx f(x) + \nabla f(x)^T (y - x) + \frac{1}{2}(y - x)^T \nabla^2 f(x)(y - x)$$

Gradient, proximal and projected gradient descent were motivated by a local quadratic expansion of f:

$$f(y) \approx f(x) + \nabla f(x)^T (y-x) + \frac{1}{2t}(y-x)^T(y-x)$$

leading to

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

Newton method improved the quadratic expansion using Hessian of f (can do projected Newton too):

$$f(y) \approx f(x) + \nabla f(x)^T (y-x) + \frac{1}{2}(y-x)^T \nabla^2 f(x)(y-x)$$

What about a simpler linear expansion of f (when does it make sense)?

$$f(y) \approx f(x) + \nabla f(x)^T (y-x)$$

Conditional gradient (Frank-Wolfe) method

Using a simpler linear expansion of f:
Choose an initial $x^{(0)} \in C$ and for $k = 1, 2, 3, \ldots$

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

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

Conditional gradient (Frank-Wolfe) method

Using a simpler linear expansion of f:
Choose an initial $x^{(0)} \in C$ and for $k = 1, 2, 3, \ldots$

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

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

The default choice for step sizes is $\gamma_k = 2/(k+1)$, $k = 1, 2, 3, \ldots$.
No dependence on Lipschitz constant, condition number, or backtracking line search parameters.

Conditional gradient (Frank-Wolfe) method

Using a simpler linear expansion of f:
Choose an initial $x^{(0)} \in C$ and for $k = 1, 2, 3, \ldots$

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

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

The default choice for step sizes is $\gamma_k = 2/(k+1)$, $k = 1, 2, 3, \ldots$. No dependence on Lipschitz constant, condition number, or backtracking line search parameters.

For any choice $0 \le \gamma_k \le 1$, we see that $x^{(k)} \in C$ by convexity. (why?)

Conditional gradient (Frank-Wolfe) method

Using a simpler linear expansion of f:
Choose an initial $x^{(0)} \in C$ and for $k = 1, 2, 3, \ldots$

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

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

Conditional gradient (Frank-Wolfe) method

Using a simpler linear expansion of f:
Choose an initial $x^{(0)} \in C$ and for $k = 1, 2, 3, \ldots$

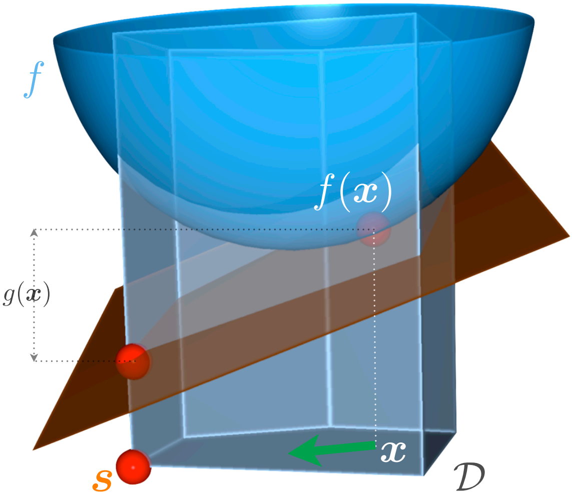
$$s^{(k-1)} \in \underset{s \in C}{\operatorname{argmin}} \ \nabla f(x^{(k-1)})^T s$$
$$x^{(k)} = (1 - \gamma_k) x^{(k-1)} + \gamma_k s^{(k-1)}$$

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

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 $C = \{x : \|x\| \le t\}$ for a norm $\|\cdot\|$? Then

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

where $\|\cdot\|_*$ is the corresponding dual norm.

Norms: $f(x) = \|x\|_p$. Let q be such that $1/p + 1/q = 1$, then

$$\|x\|_p = \max_{\|z\|_q \le 1} \ z^T x$$

And

$$\partial f(x) = \underset{\|z\|_q \le 1}{\operatorname{argmax}} \ z^T x$$

Norm constraints

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

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

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 $C = \{x : \|x\| \le t\}$. Also often simpler or cheaper than the prox operator for $\|\cdot\|$

Example: ℓ_1 regularization

For the ℓ_1-regularized problem

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

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

$$i_{k-1} \in \underset{i=1,\ldots p}{\operatorname{argmax}} \; \big|\nabla_i f(x^{(k-1)})\big|$$
$$x^{(k)} = (1-\gamma_k)x^{(k-1)} - \gamma_k t \cdot \operatorname{sign}\big(\nabla_{i_{k-1}} f(x^{(k-1)})\big) \cdot e_{i_{k-1}}$$

This is a kind of *coordinate descent.* (More on coordinate descent later.)

Note: this is a lot simpler than projection onto the ℓ_1 ball, though both require $O(n)$ operations

Example: ℓ_p regularization

For the ℓ_p-regularized problem

$$\min_x \; f(x) \;\; \text{subject to} \;\; \|x\|_p \le t$$

for $1 \le p \le \infty$, 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

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

where α 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 than projection onto the ℓ_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

$$\min_X \; f(X) \;\; \text{subject to} \;\; \|X\|_{\mathrm{tr}} \le t$$

we have $S^{(k-1)} \in -t\partial\|\nabla f(X^{(k-1)})\|_{\mathrm{op}}$. 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

$$\min_x \; f(x) \;\; \text{subject to} \;\; \|x\| \le t$$

are equivalent to those of the Lagrange problem

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

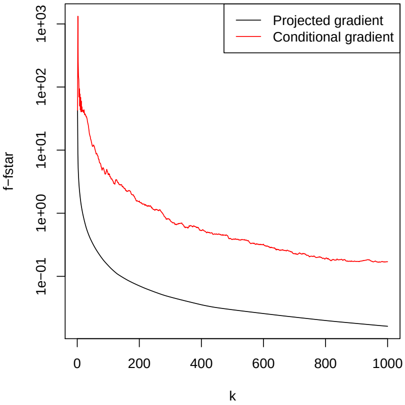
as we let the tuning parameters t and λ vary over $[0, \infty]$. More on this later.

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

- ℓ_1 norm: Frank-Wolfe update scans for maximum of gradient; proximal operator soft-thresholds the gradient step; both use $O(n)$ flops
- ℓ_p norm: Frank-Wolfe update 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)

Sub-optimality gap

Frank-Wolfe iterations admit a very natural 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

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

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

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

Rearranged, this gives the sub-optimality 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.

Convergence analysis

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

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

(Above we restrict $\gamma \in [0,1]$.) Note that $M = 0$ 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 $\gamma_k = 2/(k+1)$, $k = 1, 2, 3, \ldots$ satisfies

$$f(x^{(k)}) - f^\star \le \frac{2M}{k+2}$$

Number of iterations needed to have $f(x^{(k)}) - f^\star \le \epsilon$ is $O(1/\epsilon)$

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

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

To see this, recall that ∇f Lipschitz with constant L means

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

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

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

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:

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

Here $g(x) = \max_{s \in C} \nabla f(x)^T (x - s)$ is the sub-optimality 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

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

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 $A : \mathbb{R}^n \to \mathbb{R}^n$, define $x = Ax'$, $h(x') = f(Ax')$. Then Frank-Wolfe on $h(x')$ proceeds as

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

Multiplying by A reveals precisely the same Frank-Wolfe update as would be performed on $f(x)$. Even convergence analysis is affine invariant. Note that the curvature constant M of h is

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

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 $\gamma_k = 2/(k+1)$, $k = 1, 2, 3, \ldots$, and inaccuracy parameter $\delta \geq 0$, satisfies

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

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

Some variants

Some variants of the conditional gradient method:

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

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

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

- Fully corrective: directly update according to

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

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

References

- K. Clarkson (2010), "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 (1956), "An algorithm for quadratic programming"