Conditional Gradient Method

Frank-Wolfe

Akilesh B
Indian Institute of Technology, Hyderabad

November 19, 2016

Conditional Gradient

Consider the constrained optimization 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

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

where P_c is projection operator onto the set C. This was a special case of Proximal Gradient Descent, motivated by local quadratic expansion of f:

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

Conditional Gradient

Conditional Gradient Method also known as the Frank Wolfe method, uses a local linear expansion of f:

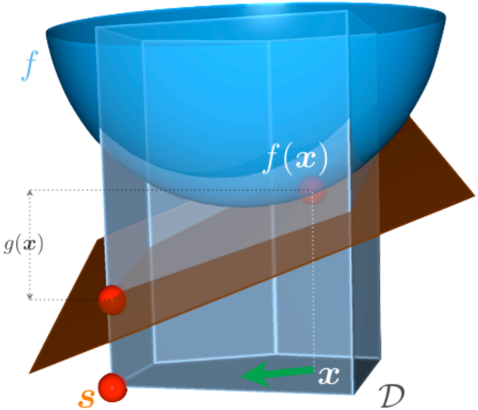
$$s^{(k-1)} \in \underset{s \in C}{\operatorname{argmin}} \nabla f(x^{(k-1)})^T s$$

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

There is no projection here, update is solved directly over constraint set C. Default choices for step size is $\gamma_k = \frac{2}{k+1}, k = 1, 2, 3 \ldots$.For any choice $0 \leq \gamma_k \leq 1$, we see that $x^{(k)} \in C$ by convexity. Updates can also be seen as

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

We are moving less and less in the direction of linearization minimizer as algorithm proceeds.



(From Jaggi 2011)

Norm Constraints

When $C = \{x : \|x\| \leq t\}$ for a norm $\|.\|$? Then

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

where $\|.\|_*$ is correspondig dual norm. Performing Frank-Wolfe steps would become very easy if we know how to compute subgradients of dual norm. This can be often simpler or cheaper than projection onto $C = \{x : \|x\| \leq t\}$ or the prox operator for $\|.\|$

Example : ℓ_1 regularization

For the ℓ_1 regularization 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

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

$$x_{(k)} = (1-\gamma_k)x^{(k-1)} - \gamma_k t.sign(\nabla_{i_{k-1}} f(x^{(k-1)}))$$

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 regularization problem :

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

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 $\frac{1}{p} + \frac{1}{q} = 1$. Note: We can choose:

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

where α is a constant such that $\|s^{(k-1)}\|_q = t$ and the Frank-Wolfe updates are usual. Note:This is a lot simpler than projection onto ℓ_p ball for general p aside from special cases (p=1,2,∞). These projections cannot be directly computed and treated as optimization problems.

Example : Trace Norm regularization

For trace-regularized problem

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

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

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

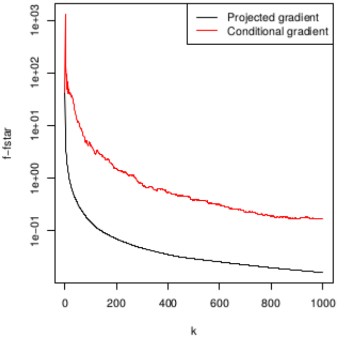
where u,v are leading left, right singluar vectors of $\nabla f(X^{(k-1)})$ and then Frank-Wolfe updates are usual. Projection onto the norm ball requires SVD.

Frank-Wolfe vs Proximal

- ℓ_1 norm: Frank-Wolfe update scans for maximum of gradient, proximal operator soft-thresholds the gradient step, both use O(n) steps.
- ℓ_p norm: Frank-Wolfe update computes raises each entry of gradient to power and sums, in O(n), proximal operator net 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 SVD.

Conditional (Not Descent) vs Projected Descent

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



Frank-Wolfe methods would converge in the same rate as 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 suboptimal gap)

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

This is an upper bound on $f(x^{(k-1)}) - f^*$.
Proof : By first order condition for convexity:

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

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

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

Rearranged, this gives duality gap above

$$\begin{aligned} f(x^{(k-1)}) - f^* &\leq \max_{s \in C} \nabla f(x^{(k-1)})^T (x^{(k-1)} - s) \\ &= \nabla f(x^{(k-1)})^T (x^{(k-1)} - s^{(k-1)}) \end{aligned}$$

Duality Gap . . .

This quantity directly comes from 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 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

$$\begin{aligned} & f(x) + f^*(u) + I_C^*(-u) \quad \text{(Fenchel Inequality)} \\ & \geq x^T u + I_C^*(-u) \end{aligned}$$

When $x = x^{(k-1)}, u = \nabla f(x^{(k-1)})$, this gives claimed gap. Duality gap can be used as stopping criterion. When it is very small, we can stop the algorithm else perform an 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 $f(y) - f(x)$ - ∇ $f(x)^T$ $(y - x)$ is called the Bregman divergence defined by f.

Theorem:

Conditional gradient method using fixed step sizes $\gamma_k = \frac{2}{k+1}$, k=1,2,3, . . . satisfies

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

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

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 $M \leq diam^2(C).L$, where

$$D = diam(C) = \max_{x,s \in C} \|x - s\|_2^2$$

To see this, recall that ∇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 $\frac{2}{\gamma^2}$,

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

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 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

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

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

Convergence proof

From previous slide we have

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

Now define $\epsilon_k = f(x^{(k)}) - f^*$, we arrive at

$$\epsilon_{k+1} \leq (1 - \gamma_k)\epsilon_k + \frac{L\gamma_k^2}{2}.D^2$$

By induction we can show that $\epsilon_k \leq \frac{2LD^2}{k+2}$. First of all, when k $= 1$, $\gamma_0 = 1$, we have

$$\epsilon_1 \leq (1 - \gamma_0)\epsilon_0 + \frac{LD^2}{2}\gamma_0^2 = \frac{LD^2}{2} \leq \frac{2}{3}LD^2$$

Now assume that it holds true for $k \geq 1$, that $\epsilon_k \leq \frac{2LD^2}{k+2}$, then

Convergence proof contd ...

$$\begin{aligned}
\epsilon_{k+1} &\leq \left(1 - \frac{2}{k+2}\right) . \frac{2LD^2}{k+2} + \frac{LD^2}{2} . \left(\frac{2}{k+2}\right)^2 \\
&= \frac{2LD^2(k+1)}{(k+2)^2} \\
&\leq \frac{2LD^2}{k+3}, \text{ since } (k+2)^2 \geq (k+1)(k+3)
\end{aligned}$$

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

Affine invariance

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

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

$$(x')^+ = (1-\gamma)x' + \gamma s'$$

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

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

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

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

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

Theorem:

Conditional gradient method using fixed step sizes $\gamma_k = \frac{2}{k+1}, k = 1, 2, 3$ $\ldots$ and inaccuracy parameter $\delta \ge 0$, satisfies

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

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

Two variants

Two important variants of the conditional gradient method:

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

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

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

- Fully corrective: directly update according to

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

  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

Take-aways

- The quadratic approximation in case of proximal descent is replaced by linear approximation in conditional gradient descent.
- Instead of first finding the minimizer and projecting back to the constraint set, this method finds minimizer over the constraint set itself.
- Sub-gradients of dual norms are easy to compute or known apriori, which makes the Frank-Wolfe iteration cheap compared to proximal or projection step.
- Frank-Wolfe is affine invariant similar to Newton method.
- Convergence rate of Frank-Wolfe is same as projected gradient descent.

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"

Questions?

Thank you!