Gradient Methods for Optimization

Stephen J. Wright[1]

[2]Computer Sciences Department,
University of Wisconsin-Madison.

Madison Summer School, July, 2018

Smooth Convex Functions

Consider $\min_{x \in \mathbb{R}^n} f(x)$, with f smooth and convex.

Usually assume $\mu I \preceq \nabla^2 f(x) \preceq LI, \ \forall_x$, with $0 \le \mu \le L$.

Thus L is a Lipschitz constant of ∇f:

$$\|\nabla f(x) - \nabla f(z)\| \le L\|x - z\|,$$

and

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

If $\mu > 0$, then f is μ-strongly convex and

$$f(y) \ge f(x) + \nabla f(x)^T (y - x) + \frac{\mu}{2}\|y - x\|_2^2.$$

Define conditioning (or condition number) as $\kappa := L/\mu$.

What's the Setup?

We consider iterative algorithms: generate $\{x_k\}$, $k = 0, 1, 2, \dots$ from

$$x_{k+1} = \Phi(x_k) \quad \text{or} \quad x_{k+1} = \Phi(x_k, x_{k-1}) \quad \text{or} \quad x_{k+1} = \Phi(x_k, x_{k-1}, \dots, x_1, x_0).$$

For now, assume we can evaluate $f(x_t)$ and $\nabla f(x_t)$ at each iteration. Some of the techniques we discuss are extendible to more general situations:

- nonsmooth f;
- f not available (or too expensive to evaluate exactly);
- only an *estimate* of the gradient is available;
- a constraint $x \in \Omega$, usually for a simple Ω (e.g. ball, box, simplex);
- nonsmooth regularization; *i.e.*, instead of simply $f(x)$, we want to minimize $f(x) + \tau\psi(x)$.

We focus on algorithms that can be adapted to those scenarios.

Steepest Descent

Minimizer x^* of f is characterized by $\nabla f(x^*) = 0$.

At a point for which $\nabla f(x) \neq 0$, can get decrease in f by moving in any direction d such that $d^T \nabla f(x) < 0$. Proof is from Taylor's theorem:

$$f(x + \alpha d) = f(x) + \alpha \nabla f(x)^T d + O(\alpha^2) < f(x), \quad \text{for } \alpha \text{ sufficiently small.}$$

Among all d with $\|d\| = 1$, the minimizer of $d^T \nabla f(x)$ is attained at $d = -\nabla f(x)$. This is the steepest descent direction.

Even when f is not convex, the direction d with $d^T \nabla f(x) = 0$ will decrease f from any point for which $\nabla f(x) \neq 0$.

Algorithms that take "reasonable" steps along $d = -\nabla f(x)$ at each iteration cannot get stuck at points $\bar{x}$ for which $\nabla f(\bar{x}) \neq 0$ — can always escape from a neighborhood of such points.

Steepest Descent

Steepest descent (a.k.a. gradient descent):

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k), \qquad \text{for some } \alpha_k > 0.$$

Different ways to select an appropriate α_k.

1. Interpolating scheme with safeguarding to identify an approximate minimizing α_k.
2. Backtrack. Try $\bar{\alpha}$, $\frac{1}{2}\bar{\alpha}$, $\frac{1}{4}\bar{\alpha}$, $\frac{1}{8}\bar{\alpha}$, ... until sufficient decrease in f.
3. Don't test for function decrease; use rules based on L and μ.
4. Set α_k based on experience with similar problems. Or adaptively.

Analysis for 1 and 2 usually yields global convergence at unspecified rate. The "greedy" strategy of getting good decrease in the current search direction may lead to better practical results.

Analysis for 3: Focuses on convergence rate, and leads to accelerated multi-step methods.

Fixed Steps

By elementary use of Taylor's theorem, and since $\nabla^2 f(x) \preceq LI$,

$$f(x_{k+1}) \le f(x_k) - \alpha_k \|\nabla f(x_k)\|^2 + \alpha_k^2 \frac{L}{2} \|\nabla f(x_k)\|_2^2.$$

For $\alpha_k \equiv 1/L$, $\quad f(x_{k+1}) \le f(x_k) - \frac{1}{2L} \|\nabla f(x_k)\|_2^2,$

thus $\quad \|\nabla f(x_k)\|^2 \le 2L[f(x_k) - f(x_{k+1})]$

Summing over first $T-1$ iterates $(k = 0, 1, \ldots, T-1)$ and telescoping the sum,

$$\sum_{k=0}^{T-1} \|\nabla f(x_k)\|^2 \le 2L[f(x_0) - f(x_T)].$$

It follows that $\nabla f(x_k) \to 0$ if f is bounded below.

Convergence Rates

From the sum above we have that

$$T \min_{k=0,1,\dots,T-1} \|\nabla f(x_k)\|^2 \le \sum_{k=0}^{T-1} \|\nabla f(x_k)\|^2 \le 2L[f(x_0) - f(x_T)],$$

and so

$$\min_{k=0,1,\dots,T-1} \|\nabla f(x_k)\| \le \sqrt{\frac{2L[f(x_0) - f(x_T)]}{T}}.$$

"Smallest gradient encountered in first T iterations shrinks like $1/\sqrt{T}$."
This result doesn't require convexity!

For convergence of function values $\{f(x_k)\}$ to their optimal value f^* in the convex case, we have the following remarkable bound:

$$f(x_T) - f^* \le \frac{L}{2T} \|x_0 - x^*\|_2^2.$$

Proof on following slides!

Proof of $1/T$ Convergence of $\{f(x_T)\}$

For any solution x^*, have

$$\begin{aligned} f(x_{k+1}) &\le f(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|^2 \\ &\le f^* + \nabla f(x_k)^T(x_k - x^*) - \frac{1}{2L}\|\nabla f(x_k)\|^2 \quad \text{(convexity)} \\ &= f(x^*) + \frac{L}{2}\left(\|x_k - x^*\|^2 - \left\|x_k - x^* - \frac{1}{L}\nabla f(x_k)\right\|^2\right) \\ &= f(x^*) + \frac{L}{2}\left(\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2\right). \end{aligned}$$

By summing over $k = 0, 1, 2, \dots, T-1$, we have

$$\begin{aligned} \sum_{k=0}^{T-1}(f(x_{k+1}) - f^*) &\le \frac{L}{2}\sum_{k=0}^{T-1}\left(\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2\right) \\ &= \frac{L}{2}\left(\|x_0 - x^*\|^2 - \|x_T - x^*\|^2\right) \le \frac{L}{2}\|x^0 - x^*\|^2. \end{aligned}$$

Continued...

Since $\{f(x^k)\}$ is nonincreasing, have

$$f(x_T) - f(x^*) \le \frac{1}{T} \sum_{k=0}^{T-1} (f(x_{k+1}) - f_\star) \le \frac{L}{2T} \|x_0 - x^*\|_2^2$$

as required. That's it!

Strongly convex: Linear Rate

From strong convexity condition, we have for any z:

$$f(z) \geq f(x_k) + \nabla f(x_k)^T (z - x_k) + \frac{\mu}{2} \|z - x_k\|^2.$$

By minimizing both sides w.r.t. z we obtain

$$f(x^*) \geq f(x_k) - \frac{1}{2\mu} \|\nabla f(x_k)\|^2,$$

so that

$$\|\nabla f(x_k)\|^2 \geq 2\mu (f(x_k) - f(x^*)). \tag{1}$$

Recall too that for step $\alpha_k \equiv 1/L$ we have

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2L} \|\nabla f(x_k)\|_2^2.$$

Subtract $f(x^*)$ from both sides of this expression and use (1):

$$(f(x_{k+1}) - f(x^*)) \leq \left(1 - \frac{\mu}{L}\right) (f(x_k) - f(x^*)).$$

A linear (geometric) rate!

A Word on Convergence Rates

Typical rates of convergence to zero for sequences such as $\{\|\nabla f(x_k)\|\}$, $\{f(x^k) - f^*\}$, and $\{\|x^k - x^*\|\}$ are

$$\begin{aligned} \phi_k &\le \frac{C_1}{\sqrt{k}}, \; \frac{C_2}{k}, \; \frac{C_3}{k^2} && \text{(sublinear)} \\ \phi_{k+1} &\le (1-c)\phi_k \;\; \text{for some } c \in (0,1) && \text{(linear)} \\ \phi_{k+1} &= o(\phi_k) && \text{(superlinear).} \end{aligned}$$

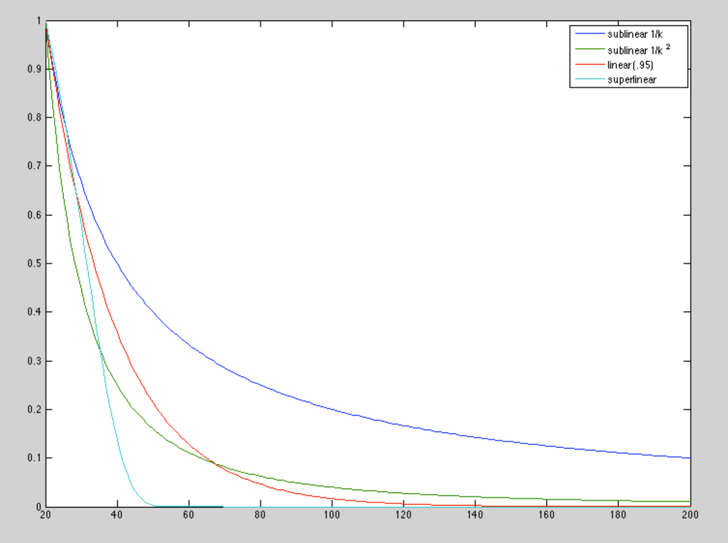
To achieve $\phi_T \le \epsilon$ for some small positive tolerance ϵ, need

$$T = O(1/\epsilon^2), \quad T = O(1/\epsilon), \quad T = O(1/\sqrt{\epsilon}) \quad \text{for sublinear rates,}$$

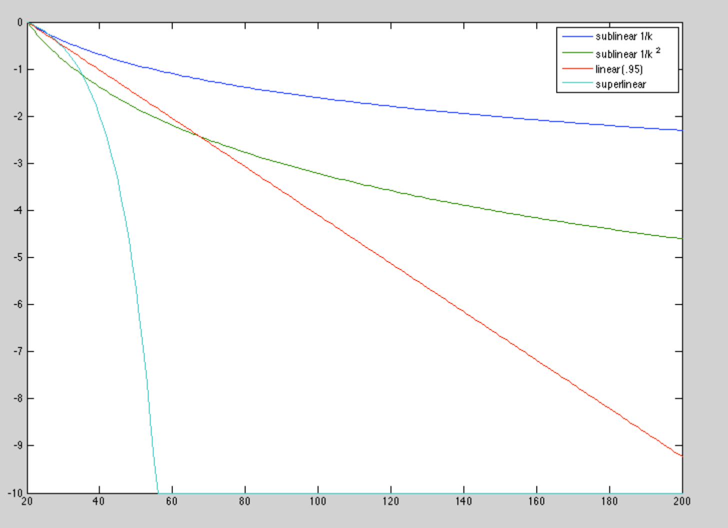
$$T = O\left(\frac{1}{c}\log\epsilon\right), \quad \text{for linear rate.}$$

Question: For a quadratic convergence rate $\phi_{k+1} \le C\phi_k^2$, how many iterations are required to obtain $\phi_T \le \epsilon$?

Convergence Rates: Standard Plots



Convergence Rates: Log Plots



Linear convergence without strong convexity

The linear convergence analysis depended on two bounds:

$$f(x_{k+1}) \le f(x_k) - a_1 \|\nabla f(x_k)\|^2, \tag{2}$$

$$\|\nabla f(x_k)\|^2 \ge a_2 (f(x_k) - f(x^*)), \tag{3}$$

for some positive a_1, a_2. In fact, many algorithms that use first derivatives, or crude estimates of first derivatives (as in stochastic gradient or coordinate descent) satisfy a bound like (2).

We derived (3) from strong convexity, but it also holds for interesting cases that are not strongly convex.

(3) is a special case of a Kurdyka-Lojasewicz (KL) property, which holds in many interesting situations — even for nonconvex f, near a local min.

More on KL

The KL property holds when f grows quadratically from its solution set:

$$f(x) - f^* \geq a_3 \operatorname{dist}(x, \text{solution set})^2, \quad \text{for some } a_3 > 0.$$

Allows nonunique solution. Proof:

$$\begin{aligned} f(x) - f^* &\leq -\nabla f(x)^T (x - x^*) \\ &\leq \|\nabla f(x)\| \|x - x^*\| \\ &\leq \|\nabla f(x)\| \sqrt{(f(x) - f^*)/a_3}. \end{aligned}$$

So obtain by rearrangement that

$$\|\nabla f(x)\|^2 \geq a_3 (f(x) - f^*).$$

KL also holds when $f(x) = \sum_{i=1}^m h(a_i^T x)$, where $h : \mathbb{R} \to \mathbb{R}$ is strongly convex, even when $m < n$, in which case $\nabla^2 f(x)$ is singular. This form of f arises in Empirical Risk Minimization (ERM).

The $1/k^2$ Speed Limit

Nesterov (2004) gives a simple example of a smooth function for which no method that generates iterates of the form $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$ can converge at a rate faster than $1/k^2$, at least for its first $n/2$ iterations.

Note that $x_{k+1} \in x_0 + \text{span}(\nabla f(x_0), \nabla f(x_1), \dots, \nabla f(x_k))$.

$$A = \begin{bmatrix} 2 & -1 & 0 & 0 & \dots & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & \dots & 0 \\ 0 & -1 & 2 & -1 & 0 & \dots & 0 \\ & & \ddots & \ddots & \ddots & & \\ 0 & \dots & & & 0 & -1 & 2 \end{bmatrix}, \qquad e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

and set $f(x) = (1/2)x^T Ax - e_1^T x$. The solution has $x^*(i) = 1 - i/(n+1)$.

If we start at $x_0 = 0$, each $\nabla f(x_k)$ has nonzeros only in its first k entries. Hence, $x_{k+1}(i) = 0$ for $i = k+1, k+2, \dots, n$. Can show that

$$f(x_k) - f^* \geq \frac{3L\|x_0 - x^*\|^2}{32(k+1)^2}.$$

Descent Directions and Line Search

Consider iteration scheme

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, 2, \dots,$$

where d_k makes an acute angle with $-\nabla f(x_k)$, that is,

$$-d_k^T \nabla f(x_k) \ge \bar{\epsilon} \|\nabla f(x_k)\| \|d_k\|. \tag{4}$$

We impose weak Wolfe conditions on steplength α_k:

$$f(x_k + \alpha d_k) \le f(x_k) + c_1 \alpha \nabla f(x_k)^T d_k, \tag{5a}$$

$$\nabla f(x_k + \alpha d_k)^T d_k \ge c_2 \nabla f(x_k)^T d_k\,. \tag{5b}$$

where $0 < c_1 < c_2 < 1$. (Typically $c_1 = .001$, $c_2 = .5$.)

- (5a) is a sufficient decrease condition;
- (5b) ensures that the step is not too short.

Second Weak Wolfe condition

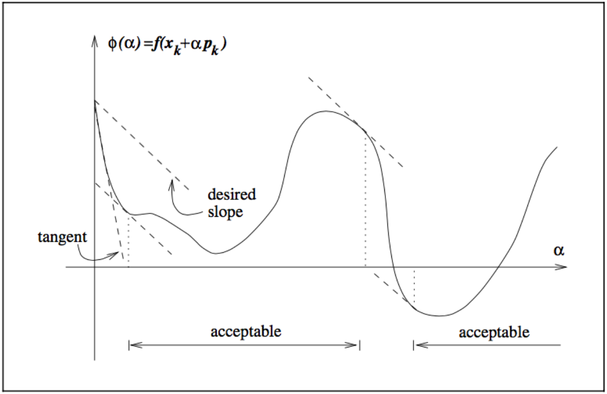


Figure 3.4 The curvature condition.

Convergence under Weak Wolfe

From condition (5b) and the Lipschitz property for ∇f, we have

$$-(1-c_2)\nabla f(x_k)^T d_k \le [\nabla f(x_k+\alpha_k d_k)-\nabla f(x_k)]^T d_k \le L\alpha_k \|d_k\|^2,$$

and thus

$$\alpha_k \ge -\frac{(1-c_2)}{L}\frac{\nabla f(x_k)^T d_k}{\|d_k\|^2}.$$

Substituting into (5a), and using (4), we have

$$\begin{aligned} f(x_{k+1}) = f(x_k+\alpha_k d_k) &\le f(x_k) + c_1\alpha_k \nabla f(x_k)^T d_k \\ &\le f(x_k) - \frac{c_1(1-c_2)}{L}\frac{(\nabla f(x_k)^T d_k)^2}{\|d_k\|^2} \\ &\le f(x_k) - \frac{c_1(1-c_2)}{L}\bar{\epsilon}^2 \|\nabla f(x_k)\|^2. \end{aligned}$$

Thus the decrease in f per iteration is a multiple of $\|\nabla f(x_k)\|^2$, just as in vanilla steepest descent with fixed steps. We thus get the same sublinear and linear convergence results.

Backtracking

Try $\alpha_k = \bar{\alpha}, \frac{\bar{\alpha}}{2}, \frac{\bar{\alpha}}{4}, \frac{\bar{\alpha}}{8}, \ldots$ until the sufficient decrease condition is satisfied.

No need to check the second Wolfe condition: the α_k thus identified is "within striking distance" of an α that's too large — so it is not too short.

Backtracking is widely used in applications, but doesn't work on nonsmooth problems, or when f is not available / too expensive.

Can show again that the decrease in f at each iteration is a multiple of $\|\nabla f(x^k)\|^2$, so the usual rates apply.

Convergence of Iterates x_k

Can we say something about the rate of convergence of $\{x_k\}$ to x^*? That is, convergence of $\|x_k - x^*\|$ or $\mathrm{dist}(x_k, \text{minimizing set})$ to zero?

In the weakly convex case, not much! $f(x^k) - f^*$ can be small while x^k is still far from x^*.

If strong convexity or quadratic growth holds, we have

$$f(x_k) - f(x^*) \geq a_3 \,\mathrm{dist}(x, \text{solution set})^2, \quad \text{for some } a_3 > 0.$$

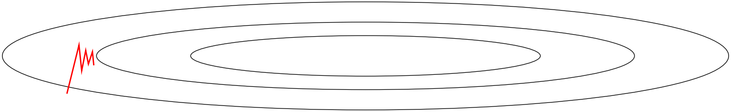
so that

$$\mathrm{dist}(x, \text{solution set}) \leq \sqrt{\frac{1}{a_3}(f(x_k) - f^*)}.$$

So we can derive convergence rates on $\mathrm{dist}(x, \text{solution set})$ from those of $f(x_k) - f^*$.

The slow linear rate is typical!

Not just a pessimistic bound! In the strongly convex case, complexity to achieve $f(x_T) - f^* \leq \epsilon(f(x_0) - f^*)$ is $O((L/m)\log\epsilon)$.



Accelerated First-Order Methods

Can we get faster rates (e.g. faster linear rates for strongly convex, faster sublinear rates for general convex) while still using only first-order information?

YES! The key idea is MOMENTUM. Search direction depends on the latest gradient $-\nabla f(x_k)$ and also on the search direction at iteration $k-1$, which encodes gradient information from all earlier iterations.

Several popular methods use momentum:

- Heavy-ball method
- Nesterov's accelerated gradient
- Conjugate gradient (linear and nonlinear).

Heavy Ball and Nesterov

Heavy Ball:

$$x_{k+1} = x_k - \alpha \nabla f(x^k) + \beta(x_k - x_{k-1}).$$

Nesterov's optimal method:

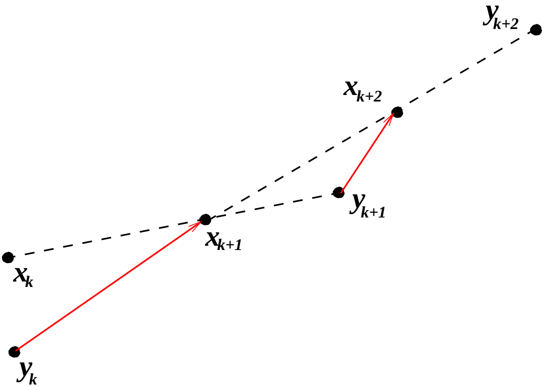
$$x_{k+1} = x_k - \alpha_k \nabla f(x_k + \beta_k(x_k - x_{k-1})) + \beta_k(x_k - x_{k-1}).$$

Typically $\alpha_k \approx 1/L$ and $\beta_k \approx 1$.

Can rewrite Nesterov by introducing an intermediate sequence $\{y_k\}$:

$$\begin{aligned} y_k &= x_k + \beta_k(x_k - x_{k-1}), \\ x_{k+1} &= y_k - \alpha_k \nabla f(y_k). \end{aligned}$$

Nesterov, illustrated



Separates the “gradient descent” and “momentum” step components.

Accelerated Gradient Convergence

Typical convergence:

$$\text{Weakly convex } \mu = 0: \quad f(x_k) - f^* = O(1/k^2);$$

$$\text{Strongly convex } \mu > 0: \quad f(x_k) - f^* \le M \left(1 - c\sqrt{\frac{\mu}{L}}\right)^k [f(x_0) - f^*],$$

for some modest positive c.

- Approach can be extended to regularized functions $f(x) + \lambda\psi(x)$: Beck and Teboulle (2009).
- Partial-gradient approaches (stochastic gradient, coordinate descent) can be accelerated in similar ways.

Heavy Ball, Quadratic

Consider heavy-ball applied to a convex quadratic:

$$f(x) = \frac{1}{2} x^T Q x,$$

where Q is symmetric positive definite with eigenvalues

$$0 < \mu = \lambda_n \le \lambda_{n-1} \le \cdots \le \lambda_2 \le \lambda_1 = L.$$

The minimizer is clearly $x^* = 0$.

Heavy ball applied to this function is

$$x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1}) = x_k - \alpha Q x_k + \beta(x_k - x_{k-1}).$$

Analyze by defining a composite iterate vector:

$$w_k := \begin{bmatrix} x_k - x^* \\ x_{k-1} - x^* \end{bmatrix} = \begin{bmatrix} x_k \\ x_{k-1} \end{bmatrix}$$

Thus

$$w_k = T w_{k-1}, \quad T := \begin{bmatrix} (1+\beta)I - \alpha Q & -\beta I \\ I & 0 \end{bmatrix}.$$

Heavy-Ball, Quadratic

Matrix T has same eigenvalues as

$$\begin{bmatrix} -\alpha\Lambda + (1+\beta)I & -\beta I \\ I & 0 \end{bmatrix}, \qquad \Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_n).$$

Can rearrange this matrix to get 2×2 blocks on the diagonal:

$$T_i := \begin{bmatrix} (1+\beta) - \alpha\lambda_i & -\beta \\ 1 & 0 \end{bmatrix}.$$

Get eigenvalues by solving quadratics:

$$u^2 - (1 + \beta - \alpha\lambda_i)u + \beta = 0,$$

Eigenvalues are all complex provided that $(1 + \beta - \alpha\lambda_i)^2 - 4\beta < 0$, which happens when

$$\beta \in \left((1 - \sqrt{\alpha\lambda_i})^2, (1 + \sqrt{\alpha\lambda_i})^2\right).$$

Heavy Ball, Quadratic

Thus the eigenvalues of T are all complex:

$$\bar{\lambda}_{i,1} = \frac{1}{2}\left[(1+\beta-\alpha\lambda_i) + i\sqrt{4\beta-(1+\beta-\alpha\lambda_i)^2}\right],$$
$$\bar{\lambda}_{i,2} = \frac{1}{2}\left[(1+\beta-\alpha\lambda_i) - i\sqrt{4\beta-(1+\beta-\alpha\lambda_i)^2}\right].$$

All eigenvalues have magnitude β!

Thus can do an eigenvalue decomposition $T = VSV^{-1}$, where S is diagonal with entries $\bar{\lambda}_{i,1}$, $\bar{\lambda}_{i,2}$, $i = 1, 2, \dots, n$.

The recurrence becomes

$$w_k = Tw_{k-1} = T^k w_0 = VS^k V^{-1} w_0.$$

Thus we have

$$\|V^{-1}w_k\| = \|S^k V^{-1} w_0\| \le \|S^k\| \|V^{-1} w_0\| = \beta^k \|V^{-1} w_0\|.$$

Note that this does not imply monotonic decrease in $\|w_k\|$, only in the scaled norm $\|V^{-1}w_k\|$.

Heavy-Ball: Optimal choice of α and β

We want to minimize β, but need β to satisfy

$$\beta \in \left((1-\sqrt{\alpha \lambda_i})^2, (1+\sqrt{\alpha \lambda_i})^2 \right), \quad \text{with } \lambda_i \in [\mu, L],$$

which is satisfied when

$$\beta = \min(|1-\sqrt{\alpha\mu}|, |1-\sqrt{\alpha L}|)^2$$

Choose α to make the two quantities on the right-hand side identical:

$$\alpha = \frac{4}{(\sqrt{L}+\sqrt{\mu})^2} \;\;\Rightarrow\;\; 1-\sqrt{\alpha\mu} = -(1-\sqrt{\alpha L}) = \frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}.$$

It follows that

$$\beta = \frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}} = 1 - \frac{2}{\sqrt{L/\mu}+1.}$$

Caution!

The heavy ball analysis is elementary and powerful.

- The asymptotic rate is slightly better than for Nesterov.
- The rate is as good as the classical conjugate gradient method for $Ax = b$.

But we need to note a few things!

- It depends on knowledge of μ and L in order to make the right choices of α and β.
- It doesn't extend neatly from quadratic to nonlinear f.
- We can't prove contraction for the weakly convex case $\mu = 0$.

Exercise: Repeat this analysis for Nesterov's optimal method (again for convex quadratic f).

Summary: Linear Convergence, Strictly Convex f

Defining $\kappa = L/\mu$, rates are approximately:

- Steepest descent: Linear rate approx $\left(1 - \frac{2}{\kappa}\right)$;
- Heavy-ball: Linear rate approx $\left(1 - \frac{2}{\sqrt{\kappa}}\right)$.

Big difference! To reduce $\|x_k - x^*\|$ by a factor ϵ, need k large enough that

$$\left(1 - \frac{2}{\kappa}\right)^k \le \epsilon \quad \Leftarrow \quad k \ge \frac{\kappa}{2}|\log \epsilon| \quad \text{(steepest descent)}$$

$$\left(1 - \frac{2}{\sqrt{\kappa}}\right)^k \le \epsilon \quad \Leftarrow \quad k \ge \frac{\sqrt{\kappa}}{2}|\log \epsilon| \quad \text{(heavy-ball)}$$

A factor of $\sqrt{\kappa}$ difference; e.g. if $\kappa = 1000$, need ~ 30 times fewer steps.

Conjugate Gradient

Basic conjugate gradient (CG) step is

$$x_{k+1} = x_k + \alpha_k p_k, \qquad p_k = -\nabla f(x_k) + \gamma_k p_{k-1}.$$

Can be identified with heavy-ball, with $\beta_k = \dfrac{\alpha_k \gamma_k}{\alpha_{k-1}}$.

However, CG can be implemented in a way that doesn't require knowledge (or estimation) of L and μ.

- Choose α_k to (approximately) miminize f along p_k;
- Choose γ_k by a variety of formulae (Fletcher-Reeves, Polak-Ribiere, etc), all of which are equivalent if f is convex quadratic. e.g.

$$\gamma_k = \frac{\|\nabla f(x_k)\|^2}{\|\nabla f(x_{k-1})\|^2}$$

Conjugate Gradient

Nonlinear CG: Variants include Fletcher-Reeves, Polak-Ribiere, Hestenes.

Restarting periodically with $p_k = -\nabla f(x_k)$ is useful (e.g. every n iterations, or when p_k is not a descent direction).

For quadratic f, convergence analysis is based on eigenvalues of A and Chebyshev polynomials, min-max arguments. Get

- Finite termination in as many iterations as there are distinct eigenvalues;
- Asymptotic linear convergence with rate approx $1 - \dfrac{2}{\sqrt{\kappa}}$. (like heavy-ball.)

(Nocedal and Wright, 2006, Chapter 5)

Nesterov's Method

Nesterov (1983) proposed a method with a nonintuitive analysis based on bounding sequences. Also analyzed by Beck and Teboulle (2009).

```
Initialize: Choose $x_0$; set $y_1 = x_0$, $t_1 = 1$;
Iterate:    $x_k \leftarrow y_k - \frac{1}{L}\nabla f(y_k)$;
            $t_{k+1} \leftarrow \frac{1}{2}\left(1 + \sqrt{1 + 4t_k^2}\right)$;
            $y_{k+1} \leftarrow x_k + \frac{t_k - 1}{t_{k+1}}(x_k - x_{k-1})$.
```

For (weakly) convex f, converges with $f(x_k) - f(x^*) \sim 1/k^2$.

When L is not known, increase an estimate of L until it's big enough.

Beck and Teboulle (2009) do the convergence analysis in 2-3 pages; elementary, but "technical." Several other more intuitive interpretations have been given recently for similar algorithms e.g. Drusvyatskiy et al. (2016).

A Non-Monotone Gradient Method: Barzilai-Borwein

Barzilai and Borwein (1988) (BB) proposed an unusual choice of α_k.
Allows f to increase (sometimes a lot) on some steps: non-monotone.

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k), \qquad \alpha_k := \arg\min_{\alpha} \|s_k - \alpha z_k\|^2,$$

where

$$s_k := x_k - x_{k-1}, \qquad z_k := \nabla f(x_k) - \nabla f(x_{k-1}).$$

Explicitly, we have

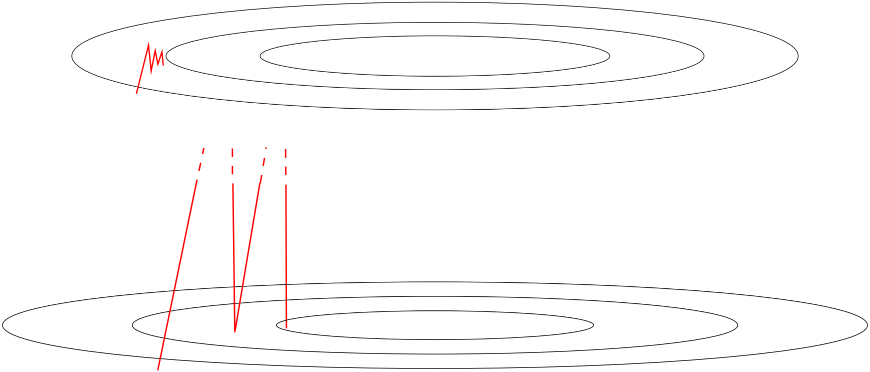
$$\alpha_k = \frac{s_k^T z_k}{z_k^T z_k}.$$

Note that for $f(x) = \frac{1}{2} x^T A x$, we have

$$\alpha_k = \frac{s_k^T A s_k}{s_k^T A^2 s_k} \in \left[\frac{1}{L}, \frac{1}{\mu} \right].$$

BB can be viewed as a quasi-Newton method, with the Hessian approximated by $\alpha_k^{-1} I$.

Comparison: BB vs Greedy Steepest Descent



There Are Many BB Variants

- use $\alpha_k = s_k^T s_k / s_k^T z_k$ in place of $\alpha_k = s_k^T z_k / z_k^T z_k$;
- alternate between these two formulae;
- hold α_k constant for a number (2, 3, 5) of successive steps;
- take α_k to be the steepest descent step from the previous iteration.

Nonmonotonicity appears essential to performance. Some variants get global convergence by requiring a sufficient decrease in f over the worst of the last M (say 10) iterates.

The original 1988 analysis in BB's paper is nonstandard and illuminating (just for a 2-variable quadratic).

In fact, most analyses of BB and related methods are nonstandard, and consider only special cases. The precursor of such analyses is Akaike (1959). More recently, see Ascher, Dai, Fletcher, Hager and others.

Extending to the Constrained Case: $x \in \Omega$

How to change these methods to handle the constraint $x \in \Omega$?
(where Ω is a closed convex set)

Some algorithms and theory stay much the same, if we can involve the constraint $x \in \Omega$ explicity in the subproblems.

Example: Nesterov's 1983 scheme requires just one calculation to be changed from the unconstrained version; see Beck and Teboulle (2009):

```
Initialize: Choose $x_0$; set $y_1 \leftarrow x_0$, $t_1 = 1$;
Iterate: $x_k \leftarrow \arg\min_{y \in \Omega} \frac{1}{2}\|y - [y_k - \frac{1}{L}\nabla f(y_k)]\|_2^2$;
    $t_{k+1} \leftarrow \frac{1}{2}\left(1 + \sqrt{1 + 4t_k^2}\right)$;
    $y_{k+1} \leftarrow x_k + \frac{t_k - 1}{t_{k+1}}(x_k - x_{k-1})$.
```

Convergence theory not affected much by the constraint.

Conditional Gradient

Also known as "Frank-Wolfe" after the authors who devised it in the 1950s. Later analysis by Dunn (around 1990). Suddenly a topic of enormous renewed interest; see for example (Jaggi, 2013).

$$\min_{x \in \Omega} f(x),$$

where f is a convex function and Ω is a closed, bounded, convex set.

Start at $x_0 \in \Omega$. At iteration k:

$$v_k := \arg\min_{v \in \Omega} v^T \nabla f(x_k);$$

$$x_{k+1} := x_k + \alpha_k (v_k - x_k), \quad \alpha_k = \frac{2}{k+2}.$$

- Potentially useful when it is easy to minimize a linear function over the *original* constraint set Ω;
- Admits an elementary convergence theory: $1/k$ sublinear rate.
- Same convergence theory holds if we use a line search for α_k.

Conditional Gradient Convergence

Diameter of Ω is $D := \max_{x,y \in \Omega} \|x - y\|$.

Theorem

Suppose that f is convex, ∇f has Lipschitz L, Ω is closed, bounded, convex with diameter D. Then conditional gradient with $\alpha_k = 2/(k+2)$ yields

$$f(x^k) - f(x^*) \le \frac{2LD^2}{k+2}, \quad k = 1, 2, \dots .$$

Proof. Setting $x = x^k$ and $y = x^{k+1} = x^k + \alpha_k (v^k - x^k)$ in the usual bound, we have

$$\begin{aligned} f(x^{k+1}) &\le f(x^k) + \alpha_k \nabla f(x^k)^T (v^k - x^k) + \frac{1}{2}\alpha_k^2 L \|v^k - x^k\|^2 \\ &\le f(x^k) + \alpha_k \nabla f(x^k)^T (v^k - x^k) + \frac{1}{2}\alpha_k^2 L D^2, \end{aligned} \tag{6}$$

where the second inequality comes from the definition of D.

Conditional Gradient Convergence, continued

For the first-order term, we have

$$\nabla f(x^k)^T (v^k - x^k) \le \nabla f(x^k)^T (x^* - x^k) \le f(x^*) - f(x^k).$$

Substitute in (6) and subtract $f(x^*)$ from both sides:

$$f(x^{k+1}) - f(x^*) \le (1 - \alpha_k)[f(x^k) - f(x^*)] + \frac{1}{2}\alpha_k^2 L D^2.$$

Now Induction. For $k = 0$, with $\alpha_0 = 1$, have

$$f(x^1) - f(x^*) \le \frac{1}{2} L D^2 < \frac{2}{3} L D^2,$$

as required. Suppose the claim holds for k, and prove for $k+1$. We have ...

$$
\begin{aligned}
f(x^{k+1}) - f(x^*) &\le \left(1 - \frac{2}{k+2}\right)[f(x^k) - f(x^*)] + \frac{1}{2}\frac{4}{(k+2)^2}LD^2 \\
&= LD^2\left[\frac{2k}{(k+2)^2} + \frac{2}{(k+2)^2}\right] \\
&= 2LD^2\frac{(k+1)}{(k+2)^2} \\
&= 2LD^2\frac{k+1}{k+2}\,\frac{1}{k+2} \\
&\le 2LD^2\frac{k+2}{k+3}\,\frac{1}{k+2} = \frac{2LD^2}{k+3},
\end{aligned}
$$

as required.

Stochastic Gradient Methods

Deal with (weakly or strongly) convex f.

- Allow f nonsmooth.
- Can't get function values $f(x)$ easily.
- At any feasible x, have access only to a cheap unbiased estimate of an element of the subgradient ∂f.

Common settings are:

$$f(x) = E_\xi F(x,\xi),$$

where ξ is a random vector with distribution P over a set Ξ. Special case:

$$f(x) = \frac{1}{m} \sum_{i=1}^{m} f_i(x),$$

where each f_i is convex and nonsmooth.
(We focus on this finite-sum formulation, but the ideas generalize.)

Applications

This setting is useful for machine learning formulations. Given data $x_i \in \mathbb{R}^n$ and labels $y_i = \pm 1$, $i = 1, 2, \ldots, m$, find w that minimizes

$$\tau \psi(w) + \frac{1}{m} \sum_{i=1}^{m} \ell(w; x_i, y_i),$$

where ψ is a regularizer, $\tau > 0$ is a parameter, and ℓ is a loss. For linear classifiers/regressors, have the specific form $\ell(w^T x_i, y_i)$.

Example: SVM with hinge loss $\ell(w^T x_i, y_i) = \max(1 - y_i(w^T x_i), 0)$ and $\psi = \| \cdot \|_1$ or $\psi = \| \cdot \|_2^2$.

Example: Logistic regression: $\ell(w^T x_i, y_i) = \log(1 + \exp(y_i w^T x_i))$. In regularized version may have $\psi(w) = \|w\|_1$.

Example: Deep Learning (the killer app!).

Subgradients

Recall: For each x in domain of f, g is a *subgradient of f at x* if

$$f(z) \geq f(x) + g^T(z-x), \qquad \text{for all } z \in \operatorname{dom} f.$$

- Right-hand side is a *supporting hyperplane*.
- The set of subgradients is called the *subdifferential*, denoted by $\partial f(x)$.
- When f is differentiable at x, have $\partial f(x) = \{\nabla f(x)\}$.

We have strong convexity with modulus $\mu > 0$ if

$$f(z) \geq f(x)+g^T(z-x)+\frac{1}{2}\mu\|z-x\|^2, \quad \text{for all } x, z \in \operatorname{dom} f \text{ with } g \in \partial f(x).$$

Generalizes the assumption $\nabla^2 f(x) \succeq \mu I$ made earlier for smooth functions.

Classical Stochastic Gradient

For the finite-sum objective, get a cheap unbiased estimate of the gradient $\nabla f(x)$ by choosing an index $i \in \{1, 2, \dots, m\}$ uniformly at random, and using $\nabla f_i(x)$ to estimate $\nabla f(x)$.

Basic SA Scheme: At iteration k, choose i_k i.i.d. uniformly at random from $\{1, 2, \dots, m\}$, choose some $\alpha_k > 0$, and set

$$x_{k+1} = x_k - \alpha_k \nabla f_{i_k}(x_k).$$

Note that x_{k+1} depends on all random indices up to iteration k, i.e. $i_{[k]} := \{i_1, i_2, \dots, i_k\}$.

When f is strongly convex, the analysis of convergence of expected square error $E(\|x_k - x^*\|^2)$ is fairly elementary — see Nemirovski et al (2009).

Define $a_k = \frac{1}{2} E(\|x_k - x^*\|^2)$. Assume there is $M > 0$ such that

$$\frac{1}{m} \sum_{i=1}^{m} \|\nabla f_i(x)\|_2^2 \leq M.$$

Rate: $1/k$

Thus

$$\begin{aligned}
&\frac{1}{2}\|x_{k+1} - x^*\|_2^2 \\
&\qquad = \frac{1}{2}\|x_k - \alpha_k \nabla f_{i_k}(x_k) - x^*\|^2 \\
&\qquad = \frac{1}{2}\|x_k - x^*\|_2^2 - \alpha_k (x_k - x^*)^T \nabla f_{i_k}(x_k) + \frac{1}{2}\alpha_k^2 \|\nabla f_{i_k}(x_k)\|^2.
\end{aligned}$$

Taking expectations, get

$$a_{k+1} \le a_k - \alpha_k E[(x_k - x^*)^T \nabla f_{i_k}(x_k)] + \frac{1}{2}\alpha_k^2 M^2.$$

For middle term, have

$$\begin{aligned}
E[(x_k - x^*)^T \nabla f_{i_k}(x_k)] &= E_{i_{[k-1]}} E_{i_k}[(x_k - x^*)^T \nabla f_{i_k}(x_k) | i_{[k-1]}] \\
&= E_{i_{[k-1]}} (x_k - x^*)^T g_k,
\end{aligned}$$

... where

$$g_k := E_{i_k}[\nabla f_{i_k}(x_k) | i_{[k-1]}] \in \partial f(x_k).$$

By strong convexity, have

$$(x_k - x^*)^T g_k \geq f(x_k) - f(x^*) + \frac{1}{2}\mu \|x_k - x^*\|^2 \geq \mu \|x_k - x^*\|^2.$$

Hence by taking expectations, we get $E[(x_k - x^*)^T g_k] \geq 2\mu a_k$. Then, substituting above, we obtain

$$a_{k+1} \leq (1 - 2\mu\alpha_k) a_k + \frac{1}{2}\alpha_k^2 M^2.$$

When

$$\alpha_k \equiv \frac{1}{k\mu},$$

a neat inductive argument (below) reveals the $1/k$ rate:

$$a_k \leq \frac{Q}{2k}, \qquad \text{for } Q := \max\left(\|x_1 - x^*\|^2, \frac{M^2}{\mu^2} \right).$$

Inductive Proof of $1/k$ Rate

Clearly true for $k = 1$. Otherwise:

$$\begin{aligned} a_{k+1} &\le (1 - 2\mu\alpha_k)a_k + \frac{1}{2}\alpha_k^2 M^2 \\ &\le \left(1 - \frac{2}{k}\right) a_k + \frac{M^2}{2k^2\mu^2} \\ &\le \left(1 - \frac{2}{k}\right) \frac{Q}{2k} + \frac{Q}{2k^2} \\ &= \frac{(k-1)}{2k^2} Q \\ &= \frac{k^2 - 1}{k^2} \frac{Q}{2(k+1)} \\ &\le \frac{Q}{2(k+1)}, \end{aligned}$$

as claimed.

But... What if we don't know μ? Or if $\mu = 0$?

The choice $\alpha_k = 1/(k\mu)$ requires strong convexity, with knowledge of the modulus μ. An underestimate of μ can greatly degrade the performance of the method (see example in Nemirovski et al. 2009).

Now describe a *Robust Stochastic Approximation* approach, which has a rate $1/\sqrt{k}$ (in function value convergence), and works for weakly convex nonsmooth functions and is not sensitive to choice of parameters in the step length.

This is the approach that generalizes to *mirror descent*, as discussed later.

Robust SA

At iteration k:

- set $x_{k+1} = x_k - \alpha_k \nabla f_{i_k}(x_k)$ as before;
- set

$$\bar{x}_k = \frac{\sum_{i=1}^k \alpha_i x_i}{\sum_{i=1}^k \alpha_i}.$$

For any $\theta > 0$, choose step lengths to be

$$\alpha_k = \frac{\theta}{M\sqrt{k}}.$$

Then $f(\bar{x}_k)$ converges to $f(x^*)$ in expectation with rate approximately $(\log k)/k^{1/2}$.

(The choice of θ is not critical.)

Analysis of Robust SA

The analysis is again elementary. As above (using i instead of k), have:

$$\alpha_i E[(x_i - x^*)^T g_i] \le a_i - a_{i+1} + \frac{1}{2}\alpha_i^2 M^2.$$

By convexity of f, and $g_i \in \partial f(x_i)$:

$$f(x^*) \ge f(x_i) + g_i^T (x^* - x_i),$$

thus

$$\alpha_i E[f(x_i) - f(x^*)] \le a_i - a_{i+1} + \frac{1}{2}\alpha_i^2 M^2,$$

so by summing iterates $i = 1, 2, \dots, k$, telescoping, and using $a_{k+1} > 0$:

$$\sum_{i=1}^{k} \alpha_i E[f(x_i) - f(x^*)] \le a_1 + \frac{1}{2} M^2 \sum_{i=1}^{k} \alpha_i^2.$$

Thus dividing by $\sum_{i=1} \alpha_i$:

$$E\left[\frac{\sum_{i=1}^k \alpha_i f(x_i)}{\sum_{i=1}^k \alpha_i} - f(x^*)\right] \le \frac{a_1 + \frac{1}{2} M^2 \sum_{i=1}^k \alpha_i^2}{\sum_{i=1}^k \alpha_i}.$$

By convexity, we have

$$f(\bar{x}_k) = f\left(\frac{\sum_{i=1}^k \alpha_i x_i}{\sum_{i=1}^k \alpha_i}\right) \le \frac{\sum_{i=1}^k \alpha_i f(x_i)}{\sum_{i=1}^k \alpha_i},$$

so obtain the fundamental bound:

$$E[f(\bar{x}_k) - f(x^*)] \le \frac{a_1 + \frac{1}{2} M^2 \sum_{i=1}^k \alpha_i^2}{\sum_{i=1}^k \alpha_i}.$$

By substituting $\alpha_i = \frac{\theta}{M\sqrt{i}}$, we obtain

$$\begin{aligned} E[f(\bar{x}_k) - f(x^*)] &\le \frac{a_1 + \frac{1}{2}\theta^2 \sum_{i=1}^k \frac{1}{i}}{\frac{\theta}{M}\sum_{i=1}^k \frac{1}{\sqrt{i}}} \\ &\le \frac{a_1 + \theta^2 \log(k+1)}{\frac{\theta}{M}\sqrt{k}} \\ &= M\left[\frac{a_1}{\theta} + \theta \log(k+1)\right]\frac{1}{\sqrt{k}}. \end{aligned}$$

That's it!

There are other variants — periodic restarting, averaging just over the recent iterates. These can be analyzed with the basic bound above.

Constant Step Size

We can also get rates of approximately $1/k$ for the strongly convex case, *without* performing iterate averaging. The tricks are to

- define the desired threshold ϵ for a_k in advance, and
- use a constant step size.

Recall the bound on a_{k+1} from a few slides back, and set $\alpha_k \equiv \alpha$:

$$a_{k+1} \le (1 - 2\mu\alpha) a_k + \frac{1}{2}\alpha^2 M^2.$$

Apply this recursively to get

$$a_k \le (1 - 2\mu\alpha)^k a_0 + \frac{\alpha M^2}{4\mu}.$$

Given $\epsilon > 0$, find α and K so that both terms on the right-hand side are less than $\epsilon/2$. The right values are:

$$\alpha := \frac{2\epsilon\mu}{M^2}, \qquad K := \frac{M^2}{4\epsilon\mu^2} \log\left(\frac{a_0}{2\epsilon}\right).$$

Constant Step Size, continued

Clearly the choice of α guarantees that the second term is less than $\epsilon/2$.

For the first term, we obtain k from an elementary argument:

$$\begin{aligned}
& (1-2\mu\alpha)^k a_0 \le \epsilon/2 \\
\Leftrightarrow \;\; & k\log(1-2\mu\alpha) \le -\log(2a_0/\epsilon) \\
\Leftarrow \;\; & k(-2\mu\alpha) \le -\log(2a_0/\epsilon) \qquad \text{since } \log(1+x) \le x \\
\Leftrightarrow \;\; & k \ge \frac{1}{2\mu\alpha}\log(2a_0/\epsilon),
\end{aligned}$$

from which the result follows, by substituting for α in the right-hand side.

If μ is underestimated by a factor of β, we undervalue α by the same factor, and K increases by $1/\beta$. (Easy modification of the analysis above.)

Thus, underestimating μ gives a mild performance penalty.

Constant Step Size: Summary

PRO: Avoid averaging, $1/k$ sublinear convergence, insensitive to underestimates of μ.

CON: Need to estimate probably unknown quantities: besides μ, we need M (to get α) and a_0 (to get K).

We use constant size size in the parallel SG approach HOGWILD!, *to be described later.*

But the step is chosen by trying different options and seeing which seems to be converging fastest. We don't actually try to estimate all the quantities in the theory and construct α that way.

Mirror Descent

The step from x_k to x_{k+1} can be viewed as the solution of a subproblem:

$$x_{k+1} = \arg\min_z \nabla f_{i_k}(x_k)^T (z - x_k) + \frac{1}{2\alpha_k} \|z - x_k\|_2^2,$$

a linear estimate of f plus a prox-term. This provides a route to handling constrained problems, regularized problems, alternative prox-functions.

For the constrained problem $\min_{x \in \Omega} f(x)$, simply add the restriction $z \in \Omega$ to the subproblem above.

We may use other prox-functions in place of $(1/2)\|z - x\|_2^2$ above. Such alternatives may be particularly well suited to particular constraint sets Ω.

Mirror Descent is the term used for such generalizations of the SA approaches above.

Mirror Descent cont'd

Given constraint set Ω, choose a norm $\|\cdot\|$ (not necessarily Euclidean). Define the *distance-generating function* ω to be a strongly convex function on Ω with modulus 1 with respect to $\|\cdot\|$, that is,

$$(\omega'(x) - \omega'(z))^T (x - z) \geq \|x - z\|^2, \quad \text{for all } x, z \in \Omega,$$

where $\omega'(\cdot)$ denotes an element of the subdifferential.

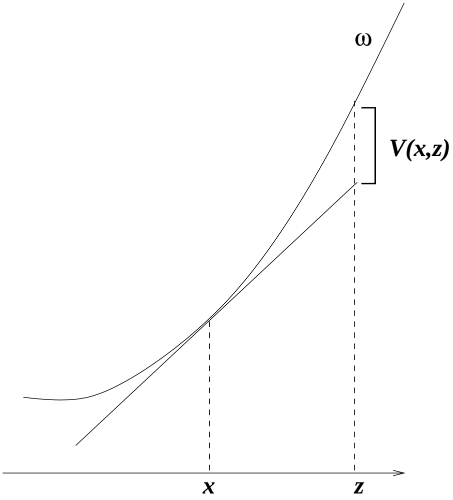
Now define the *prox-function* $V(x, z)$ as follows:

$$V(x, z) = \omega(z) - \omega(x) - \omega'(x)^T (z - x).$$

This is also known as the *Bregman distance*. We can use it in the subproblem in place of $\frac{1}{2}\|\cdot\|^2$:

$$x_{k+1} = \arg\min_{z \in \Omega} \nabla f_{i_k}(x_k)^T (z - x_k) + \frac{1}{\alpha_k} V(z, x_k).$$

Bregman distance is the deviation of ω from linearity:



Bregman Distances: Examples

For *any* Ω, we can use $\omega(x) := (1/2)\|x - \bar{x}\|_2^2$, leading to the "universal" prox-function

$$V(x,z) = (1/2)\|x - z\|_2^2$$

For the simplex

$$\Omega = \{x \in \mathbb{R}^n \,:\, x \geq 0,\ \sum_{i=1}^n x_i = 1\},$$

we can use instead the 1-norm $\|\cdot\|_1$, choose ω to be the entropy function

$$\omega(x) = \sum_{i=1}^n x_i \log x_i,$$

leading to Bregman distance (Kullback-Liebler divergence)

$$V(x,z) = \sum_{i=1}^n z_i \log(z_i/x_i),$$

which is standard measure of distance between two probability

Minibatching

$$f(x) = \frac{1}{m} \sum_{i=1}^{m} f_i(x),$$

can clump the f_i into disjoint "minibatches." Then write

$$f(x) = \frac{1}{b} \sum_{j=1}^{b} f_{[j]}(x),$$

where

$$f_{[j]}(x) = \sum_{i \in \mathcal{B}_j} f_i(x),$$

and $\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_b$ is a partition of $\{1, 2, \dots, n\}$.

Then apply SG to the batched form. Advantages:

- bigger chunks of work may make for more efficient implementation.
- lower variance in estimate of gradient $\nabla f(x)$ by $\nabla f_{[j]}(x)$.
- allows parallelism — the work of evaluating $\nabla f_i(x)$ for $i \in \mathcal{B}_j$ can be farmed out to various processors or cores.

Parallel Stochastic Gradient

Several approaches tried, for $f(x) = \sum_{i=1}^m f_i(x)$.

- **Asynchronous:** HOGWILD!**:** Each core grabs the centrally-stored x and evaluates $\nabla f_i(x)$ for some random i, then writes the updates back into x (Niu, Ré, Recht, Wright, NIPS, 2011).
- **Synchronous:** "Internal" parallelism in evaluation of gradient $\nabla f_{[j]}(x)$ for minibatch $\mathcal{B}_j$. (No changes needed to analysis.)

HOGWILD!: Each processor runs independently:

1. Sample i uniformly from $\{1, 2, \dots, m\}$;
2. Read current state of x and evaluate $g_i = \nabla f_i(x)$;
3. Update $x \leftarrow x - \alpha g_i$;

HOGWILD! Convergence

Analysis of asynchronous methods is nontrivial

- Updates can be old by the time they are applied, but we assume a bound τ on their age. This value τ is related to the number of cores involved in the computation.
- Processors can overwrite each other's work, but this effect is less important if each f_i depends on just a few components of x

Analysis of Niu et al (2011) simplified / generalized by Richtarik (2012) and many others.

Essentially show that similar $1/k$ rate to serial SG are possible provided that the maximum age τ of the updates is not too large.

Coordinate Descent

Consider unconstrained smooth problem: min $f(x)$. In coordinate descent (CD), we take a step in one component of x at a time.

```
Set Choose $x^1 \in \mathbb{R}^n$;
for $\ell = 0, 1, 2, \ldots$ do
    for $j = 1, 2, \ldots, n$ do
        Define $k = \ell n + j$;
        Choose index $i = i(\ell, j) \in \{1, 2, \ldots, n\}$;
        Choose $\alpha_k > 0$;
        $x^{k+1} \leftarrow x^k - \alpha_k \nabla_i f(x^k) e_i$;
    end for
end for
```

Here $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)^T$, with the 1 in position i.

Each ℓ represents one "epoch" (batch of n CD steps). Each j is an inner iteration.

Allows generalizations to *blocks* (multiple components), separable regularization terms.

Economics of CD

CD is most useful when the cost of one step is about $1/n$ of the cost of a full gradient step. These "economics" hold true in some important classes of problems:

- Empirical Risk Minimization (ERM):

  $$f(x) = \frac{1}{m} \sum_{j=1}^{m} h_j(A_{j\cdot} x) + \lambda \sum_{i=1}^{n} \Omega_i(x_i),$$

  where $A_{j\cdot}$ is j-th row of an $m \times n$ matrix A. Examples: Least squares, logistic regression, support vector machines.
- Graph-based objectives:

  $$f(x) = \sum_{(i,j)\in E} f_{ij}(x_i, x_j),$$

  where each variable x_i is associated with a node in the graph and E is the set of edges.

On certain problems, CD can be faster than full-gradient steepest descent — up to n times faster.

CD Variants

One way to distinguish between different variants of CD is by the order in which components are considered. That is, the choice of $i(\ell, j)$ at epoch ℓ, inner iteration j.

- CCD (Cyclic CD): $i(\ell, j) = j$.
- RCD (Randomized CD a.k.a. Stochastic CD): $i(\ell, j)$ is chosen uniformly at random from $\{1, 2, \dots, n\}$ — sampling-with-replacement.
- RPCD (Random-Permutations Cyclic CD): At the start of epoch ℓ, we choose a random permutation of $\{1, 2, \dots, n\}$, denoted by π_ℓ. Index $i(\ell, j)$ is chosen to be the jth entry in π_ℓ. This is sampling without replacement within each cycle.

Choice of α_k is also important — exact line search, or some other, looser step along $\nabla_i f(x^k)$.

Important Quantities

We assume convex f. Certain global properties are important for convergence analysis.

$L_{\max}$ is a componentwise Lipschitz constant for ∇f:

$$|\nabla_i f(x + te_i) - \nabla_i f(x)| \le L_i |t|, \quad L_{\max} = \max_{i=1,2,\dots,n} L_i.$$

L is the Lipschitz constant for the whole gradient:

$$\|\nabla f(y) - \nabla f(z)\| \le L \|y - z\|.$$

On quadratic functions, we have

$$1 \le \frac{L}{L_{\max}} \le n.$$

The max is achieved by $f(x) = (e^T x)^2$, where $e = (1, 1, \dots, 1)^T$

RCD Convergence

Convergence of sampling-with-replacement randomized with $\alpha_k = 1/L_{\max}$.

$$\begin{aligned}
f(x^{k+1}) &= f\left(x^k - \alpha_k \nabla_{i_k} f(x^k) e_{i_k}\right) \\
&\le f(x^k) - \alpha_k [\nabla_{i_k} f(x^k)]^2 + \frac{1}{2}\alpha_k^2 L_{i_k} [\nabla_{i_k} f(x^k)]^2 \\
&\le f(x^k) - \alpha_k \left(1 - \frac{L_{\max}}{2}\alpha_k\right) [\nabla_{i_k} f(x^k)]^2 \\
&= f(x^k) - \frac{1}{2L_{\max}} [\nabla_{i_k} f(x^k)]^2,
\end{aligned}$$

where we used $\alpha_k = 1/L_{\max}$. Take expectations w.r.t. i_k (note that x^k does not depend on i_k), get

$$\begin{aligned}
E_{i_k}[f(x^{k+1})] &\le f(x^k) - \frac{1}{2L_{\max}} \frac{1}{n} \sum_{i=1}^{n} [\nabla_i f(x^k)]^2 \\
&= f(x^k) - \frac{1}{2nL_{\max}} \|\nabla f(x^k)\|^2.
\end{aligned}$$

Subtract f^* from both sides and take expectation w.r.t $i_0, i_1, \ldots, i_{k-1}$:

$$\phi_{k+1} \le \phi_k - \frac{1}{2nL_{\max}} E\left(\|\nabla f(x^k)\|^2\right).$$

If f is strongly convex with modulus μ, we have

$$f(y) \ge f(x) + \nabla f(x)^T(y-x) + \frac{\mu}{2}\|y-x\|^2.$$

Fixing $x = x^k$ and minimizing both sides of this expression (separately) w.r.t. y, obtain

$$f^* \ge f(x^k) - \frac{1}{2\mu}\|\nabla f(x^k)\|^2$$

so by rearranging and taking expectations:

$$E\left(\|\nabla f(x^k)\|^2\right) \ge 2\mu E[f(x^k) - f^*] = 2\mu\phi_k.$$

Thus get linear convergence in expectation:

$$\phi_{k+1} \le \left(1 - \frac{\mu}{nL_{\max}}\right)\phi_k, \quad k = 0, 1, 2, \ldots .$$

Nesterov (2012)

Comparison with Steepest Descent

This rate is for a single CD step. Over an epoch of n steps, we get a decrease of

$$\left(1 - \frac{\mu}{nL_{\max}}\right)^n \approx \left(1 - \frac{\mu}{L_{\max}}\right).$$

Compare this with the decrease factor for one step of steepest descent (SD) with a fixed step $\alpha_k = 1/L$ (analyzed earlier):

$$\left(1 - \frac{\mu}{L}\right).$$

When $L_{\max} \ll L$, RCD can be a lot faster than SD!

To get convergence to $\phi_K \leq \epsilon$, need approximately

$$\begin{aligned} \frac{L}{\mu}|\log \epsilon| & \quad \text{SD iterations} \\ \frac{L_{\max}}{\mu}|\log \epsilon| & \quad \text{RCD epochs.} \end{aligned}$$

Cyclic CD Convergence

Convergence rate results for CCD are somewhat weaker (Beck and Tetruashvili (2013)). Depending on choice of steplength, the worst-case bounds are generally worse than for RCD or SD.

$$
\begin{aligned}
\text{CCD with } \alpha = 1/L_{\max} : &\quad \frac{2nL^2/L_{\max}}{\mu} |\log \epsilon| \\
\text{CCD with } \alpha = 1/L : &\quad \frac{2Ln}{\mu} |\log \epsilon| \\
\text{CCD with } \alpha = 1/(\sqrt{n}L) : &\quad \frac{4L\sqrt{n}}{\mu} |\log \epsilon|.
\end{aligned}
$$

- "Worse" than the expected linear rate for RCD by factors of $\sqrt{n}L/L_{\max}$ to $nL^2/L_{\max}^2$;
- "Worse" than linear rate for SD by factors of $\sqrt{n}$ to $nL/L_{\max}$.

CD Convergence

CCD can achieve this worst-case behavior for a particular quadratic function, with Hessian

$A = (1-\delta)ee^T + \delta I, \quad$ where $e = (1, 1, \ldots, 1)^T$ and δ is small and positive.

(see Sun and Ye (2016).)

Random-Permutations Cyclic (RPCD) behaves provably well on this example (Lee and Wright (2016)) — better than RCD.

In general RPCD works as well (or better than) RCD, but it is difficult to analyze in general.

Still, on many problems, when $L/L_{\max}$ is moderate, there is not much difference in performance between SD and all variants of CD.

References I

Akaike, H. (1959). On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method. *Annals of the Institute of Statistical Mathematics*, 11(1):1–16.

Barzilai, J. and Borwein, J. M. (1988). Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8:141–148.

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-threshold algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202.

Beck, A. and Tetruashvili, L. (2013). On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization*, 23(4):2037–2060.

Drusvyatskiy, D., Fazel, M., and Roy, S. (2016). An optimal first-order method based on optimal quadratic averaging. Technical Report arXiv:1604.06543, University of Washington.

Jaggi, M. (2013). Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*.

Lee, C.-p. and Wright, S. J. (2016). Random permutations fix a worst case for cyclic coordinate descent. Technical Report arXiv:1607.08320, Computer Sciences Department, University of Wisconsin-Madison.

Nesterov, Y. (1983). A method for unconstrained convex problem with the rate of convergence $O(1/k^2)$. *Doklady AN SSSR*, 269:543–547.

Nesterov, Y. (2012). Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22:341–362.

References II

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, New York, second edition.

Sun, R. and Ye, Y. (2016). Worst-case complexity of cyclic coordinate descent: $o(n^2)$ gap with randomized version. Technical Report arXiv:1604.07130, Department of Management Science and Engineering, Stanford University, Stanford, California.

Wright, S. J. (2015). Coordinate descent algorithms. *Mathematical Programming, Series B*, 151:3–34.