Some Relevant Topics in Optimization

Stephen Wright

University of Wisconsin-Madison

IPAM, July 2012

1. Introduction/Overview
2. Gradient Methods
3. Stochastic Gradient Methods for Convex Minimization
4. Sparse and Regularized Optimization
5. Decomposition Methods
6. Augmented Lagrangian Methods and Splitting

Introduction

Learning from data leads naturally to optimization formulations. Typical ingredients of a learning problem include

- Collection of "training" data, from which we want to learn to make inferences about future data.
- Parametrized model, whose parameters can in principle be determined from training data + prior knowledge.
- Objective that captures prediction errors on the training data and deviation from prior knowledge or desirable structure.

Other typical properties of learning problems are huge underlying data set, and requirement for solutions with only low-medium accuracy.

Formulation as an optimization problem can be difficult and controversial. However there are several important paradigms in which the issue is well settled. (e.g. Support Vector Machines, Logistic Regression, Recommender Systems.)

Optimization Formulations

There is a wide variety of optimization formulations for machine learning problems. But several common issues and structures arise in many cases.

- Imposing structure. Can include regularization functions in the objective or constraints.
  - $\|x\|_1$ to induce sparsity in the vector x;
  - Nuclear norm $\|X\|_*$ (sum of singular values) to induce low rank in X.
- Objective: Can be derived from Bayesian statistics + maximum likelihood criterion. Can incorporate prior knowledge.

Objectives f have distinctive properties in several applications:

- Partially separable: $f(x) = \sum_{e \in E} f_e(x_e)$, where each x_e is a subvector of x, and each term f_e corresponds to a single item of data.
- Sometimes possible to compute subvectors of the gradient ∇f at proportionately lower cost than the full gradient.
- These two properties are often combined: In partially separable f, subvector x_e is often small.

Examples: Partially Separable Structure

1. SVM with hinge loss:

$$f(w) = C \sum_{i=1}^{N} \max(1 - y_i(w^T x_i), 0) + \frac{1}{2}\|w\|^2,$$

where variable vector w contains feature weights, x_i are feature vectors, $y_i = \pm 1$ are labels, and $C > 0$ is a parameter.

2. Matrix completion. Given $k \times n$ marix M with entries $(u, v) \in E$ specified, seek L $(k \times r)$ and R $(n \times r)$ such that $M \approx LR^T$.

$$\min_{L,R} \sum_{(u,v)\in E} \left\{ (L_{u\cdot} R_{v\cdot}^T - M_{uv})^2 + \mu_u \|L_{u\cdot}\|_F^2 + \mu_v \|R_{v\cdot}\|_F^2 \right\}.$$

Examples: Partially Separable Structure

3. Regularized logistic regression (2 classes):

$$f(w) = -\frac{1}{N}\sum_{i=1}^{N} \log(1 + \exp(y_i w^T x_i)) + \mu\|w\|_1.$$

4. Logistic regression (M classes): $y_{ij} = 1$ if data point i is in class j; $y_{ij} = 0$ otherwise. $w_{[j]}$ is the subvector of w for class j.

$$f(w) = -\frac{1}{N}\sum_{i=1}^{N}\left[\sum_{j=1}^{M} y_{ij}(w_{[j]}^T x_i) - \log(\sum_{j=1}^{M}\exp(w_{[j]}^T x_i))\right] + \sum_{j=1}^{M}\|w_{[j]}\|_2^2.$$

Examples: "Partial Gradient" Structure

1. Dual, nonlinear SVM:

$$\min_{\alpha} \frac{1}{2}\alpha^T K\alpha - \mathbf{1}^T\alpha \;\text{ s.t. } 0 \le \alpha \le C\mathbf{1},\; y^T\alpha = 0,$$

   where $K_{ij} = y_i y_j k(x_i, x_j)$, with $k(\cdot,\cdot)$ a kernel function. Subvectors of the gradient $K\alpha - \mathbf{1}$ can be updated and maintained economically.

2. Logistic regression (again): Gradient of log-likelihood function is

$$\frac{1}{N}X^T u, \;\text{ where } u_i = -y_i(1 + \exp(y_i w^T x_i)),\; i = 1, 2, \ldots, N.$$

   If w is sparse, it may be cheap to evaluate u, which is dense. Then, evaluation of partial gradient $[\nabla f(x)]_{\mathcal{G}}$ may be cheap.

Partitioning of x may also arise naturally from problem structure, parallel implementation, or administrative reasons (e.g. decentralized control).

(Block) Coordinate Descent methods that exploit this property have been successful. (More tomorrow.)

Batch vs Incremental

Considering the partially separable form

$$f(x) = \sum_{e \in E} f_e(x_e),$$

the size $|E|$ of the training set can be very large. Practical considerations, and differing requirements for solution accuracy lead to a fundamental divide in algorithmic strategy.

Incremental: Select a single e at random, evaluate $\nabla f_e(x_e)$, and take a step in this direction. (Note that $E(\nabla f_e(x_e)) = |E|^{-1} \nabla f(x)$.) *Stochastic Approximation (SA).*

Batch: Select a subset of data $\tilde{E} \subset E$, and minimize the function $\tilde{f}(x) = \sum_{e \in \tilde{E}} f_e(x_e)$. *Sample-Average Approximation (SAA).*

Minibatch is a kind of compromise: Aggregate the e into small groups, consisting of 10 or 100 individual terms, and apply incremental algorithms to the redefined summation. (Gives lower-variance gradient estimates.)

Background: Optimization and Machine Learning

A long history of connections. Examples:

- Back-propagation for neural networks was recognized in the 80s or earlier as an incremental gradient method.
- Support Vector machine formulated as a linear and quadratic program in the late 1980s. Duality allowed formulation of nonlinear SVM as a convex QP. From late 1990s, many specialized optimization methods were applied: interior-point, coordinate descent / decomposition, cutting-plane, stochastic gradient.
- Stochastic gradient. Originally Robbins-Munro (1951). Optimizers in Russia developed algorithms from 1980 onwards. Rediscovered by machine learning community around 2004 (Bottou, LeCun). Parallel and independent work in ML and Optimization communities until 2009. Intense research continues.

Connections are now stronger than ever, with much collaborative and crossover activity.

Gradient Methods

$\min f(x)$, with smooth convex f. Usually assume

$$\mu I \preceq \nabla^2 f(x) \preceq LI \text{ for all } x,$$

with $0 \leq \mu \leq L$. (L is thus a Lipschitz constant on the gradient ∇f.)

$\mu > 0 \Rightarrow$ strongly convex. Have

$$f(y) - f(x) - \nabla f(x)^T (y - x) \geq \frac{1}{2}\mu \|y - x\|^2.$$

(Mostly assume $\|\cdot\| := \|\cdot\|_2$.) Define conditioning $\kappa := L/\mu$.

Sometimes discuss convex quadratic f:

$$f(x) = \frac{1}{2} x^T A x, \text{ where } \mu I \preceq A \preceq LI.$$

What's the Setup?

Assume in this part of talk that we can evaluate f and ∇f at each iterate x_j. But we are interested in extending to broader class of problems:

- nonsmooth f;
- f not available;
- only an *estimate* of the gradient (or subgradient) is available;
- impose a constraint $x \in \Omega$ for some simple set Ω (e.g. ball, box, simplex);
- a nonsmooth regularization term may be added to the objective f.

Focus on algorithms that can be adapted to these circumstances.

Steepest Descent

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

Different ways to identify an appropriate α_k.

1. Hard: Interpolating scheme with safeguarding to identify an approximate minimizing α_k.
2. Easy: Backtracking. $\bar{\alpha}$, $\frac{1}{2}\bar{\alpha}$, $\frac{1}{4}\bar{\alpha}$, $\frac{1}{8}\bar{\alpha}$, ... until a sufficient decrease in f is obtained.
3. Trivial: Don't test for function decrease. Use rules based on L and μ.

Traditional analysis for 1 and 2: Usually yields global convergence at unspecified rate. The "greedy" strategy of getting good decrease from the current search direction is appealing, and may lead to better practical results.

Analysis for 3: Focuses on convergence rate, and leads to accelerated multistep methods.

Line Search

Works for nonconvex f also.

Seek α_k that satisfies Wolfe conditions: "sufficient decrease" in f:

$$f(x_k - \alpha_k \nabla f(x_k)) \leq f(x_k) - c_1 \alpha_k \|\nabla f(x_k)\|^2, \qquad (0 < c_1 \ll 1)$$

while not being too small (significant increase in the directional derivative):

$$\nabla f(x_{k+1})^T \nabla f(x_k) \geq -c_2 \|\nabla f(x_k)\|^2, \qquad (c_1 < c_2 < 1).$$

Can show that for convex f, accumulation points $\bar{x}$ of $\{x_k\}$ are stationary: $\nabla f(\bar{x}) = 0$. (Optimal, when f is convex.)

Can do a one-dimensional line search for α_k, taking minima of quadratic or cubics that interpolate the function and gradient information at the last two values tried. Use brackets to ensure steady convergence. Often find a suitable α within 3 attempts.

(See e.g. Ch. 3 of Nocedal & Wright, 2006)

Backtracking

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

(No need to check the second Wolfe condition, as the value of α_k thus identified is "within striking distance" of a value that's too large — so it is not too short.)

These methods are widely used in many applications, but they don't work on nonsmooth problems when subgradients replace gradients, or when f is not available.

Constant (Short) Steplength

By elementary use of Taylor's theorem, obtain

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

For $\alpha_k \equiv 1/L$, have

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

thus

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

By summing from $k = 0$ to $k = N$, and telescoping the sum, we have

$$\sum_{k=1}^{N} \|\nabla f(x_k)\|^2 \leq 2L[f(x_0) - f(x_{N+1})].$$

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

Rate Analysis

Another elementary use of Taylor's theorem shows that

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \alpha_k(2/L - \alpha_k)\|\nabla f(x_k)\|^2,$$

so that $\{\|x_k - x^*\|\}$ is decreasing.

Define for convenience: $\Delta_k := f(x_k) - f(x^*)$.

By convexity, have

$$\Delta_k \leq \nabla f(x_k)^T(x_k - x^*) \leq \|\nabla f(x_k)\| \, \|x_k - x^*\| \leq \|\nabla f(x_k)\| \, \|x_0 - x^*\|.$$

From previous page (subtracting $f(x^*)$ from both sides of the inequality), and using the inequality above, we have

$$\Delta_{k+1} \leq \Delta_k - (1/2L)\|\nabla f(x_k)\|^2 \leq \Delta_k - \frac{1}{2L\|x_0 - x^*\|^2}\Delta_k^2.$$

Weakly convex: $1/k$ sublinear; Strongly convex: linear

Take reciprocal of both sides and manipulate (using $(1-\epsilon)^{-1} \geq 1+\epsilon$):

$$\frac{1}{\Delta_{k+1}} \geq \frac{1}{\Delta_k} + \frac{1}{2L\|x_0 - x^*\|^2} \geq \frac{1}{\Delta_0} + \frac{k+1}{2L\|x_0 - x^*\|^2},$$

which yields

$$f(x_{k+1}) - f(x^*) \leq \frac{2L\|x_0 - x\|^2}{k+1}.$$

The classic $1/k$ convergence rate!

By assuming $\mu > 0$, can set $\alpha_k \equiv 2/(\mu + L)$ and get a linear (geometric) rate: Much better than sublinear, in the long run

$$\|x_k - x^*\|^2 \leq \left(\frac{L-\mu}{L+\mu}\right)^{2k} \|x_0 - x^*\|^2 = \left(1 - \frac{2}{\kappa+1}\right)^{2k} \|x_0 - x^*\|^2.$$

Since by Taylor's theorem we have

$$\Delta_k = f(x_k) - f(x^*) \leq (L/2)\|x_k - x^*\|^2,$$

it follows immediately that

$$f(x_k) - f(x^*) \leq \frac{L}{2}\left(1 - \frac{2}{\kappa + 1}\right)^{2k} \|x_0 - x^*\|^2.$$

Note: A geometric / linear rate is generally much better than any sublinear ($1/k$ or $1/k^2$) rate.

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

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

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 $i = k+1, k+2, \ldots, n$. Can show

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

Exact minimizing α_k: Faster rate?

Take α_k to be the exact minimizer of f along $-\nabla f(x_k)$. Does this yield a better rate of linear convergence?

Consider the convex quadratic $f(x) = (1/2)x^T Ax$. (Thus $x^* = 0$ and $f(x^*) = 0$.) Here κ is the condition number of A.
We have $\nabla f(x_k) = Ax_k$. Exact minimizing α_k:

$$\alpha_k = \frac{x_k^T A^2 x_k}{x_k^T A^3 x_k} = \arg\min_{\alpha} \frac{1}{2}(x_k - \alpha A x_k)^T A (x_k - \alpha A x_k),$$

which is in the interval $\left[\frac{1}{L}, \frac{1}{\mu}\right]$. Thus

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2}\frac{(x_k^T A^2 x_k)^2}{(x_k^T A x_k)(x_k^T A^3 x_k)},$$

so, defining $z_k := Ax_k$, we have

$$\frac{f(x_{k+1}) - f(x^*)}{f(x_k) - f(x^*)} \leq 1 - \frac{\|z_k\|^4}{(z_k^T A^{-1} z_k)(z_k^T A z_k)}.$$

Use Kantorovich inequality:

$$(z^T Az)(z^T A^{-1} z) \leq \frac{(L+\mu)^2}{4L\mu}\|z\|^4.$$

Thus

$$\frac{f(x_{k+1}) - f(x^*)}{f(x_k) - f(x^*)} \leq 1 - \frac{4L\mu}{(L+\mu)^2} = \left(1 - \frac{2}{\kappa+1}\right)^2,$$

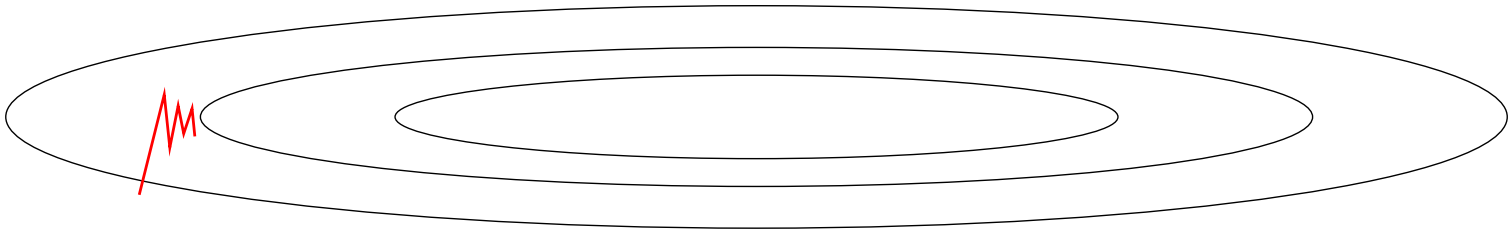
and so

$$f(x_k) - f(x^*) \leq \left(1 - \frac{2}{\kappa+1}\right)^{2k} [f(x_0) - f(x^*)].$$

No improvement in the linear rate over constant steplength.

The slow linear rate is typical!

Not just a pessimistic bound!



Multistep Methods: Heavy-Ball

Enhance the search direction by including a contribution from the *previous* step.

Consider first constant step lengths:

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

Analyze by defining a composite iterate vector:

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

Thus

$$w_{k+1} = B w_k + o(\|w_k\|), \qquad B := \begin{bmatrix} -\alpha \nabla^2 f(x^*) + (1+\beta) I & -\beta I \\ I & 0 \end{bmatrix}.$$

B has same eigenvalues as

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

where λ_i are the eigenvalues of $\nabla^2 f(x^*)$. Choose α, β to explicitly minimize the max eigenvalue of B, obtain

$$\alpha = \frac{4}{L}\frac{1}{(1+1/\sqrt{\kappa})^2}, \quad \beta = \left(1 - \frac{2}{\sqrt{\kappa}+1}\right)^2.$$

Leads to linear convergence for $\|x_k - x^*\|$ with rate approximately

$$\left(1 - \frac{2}{\sqrt{\kappa}+1}\right).$$

Summary: Linear Convergence, Strictly Convex f

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

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

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

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

A factor of $\sqrt{\kappa}$ difference. e.g. if $\kappa = 100$, need 10 times fewer steps.

Conjugate Gradient

Basic step is

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

We can identify it with heavy-ball by setting $\beta_k = \alpha_k \gamma_k / \alpha_{k-1}$. 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.

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

CG, cont'd.

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

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

For f quadratic, 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 $1 - 2/\sqrt{\kappa}$. (Like heavy-ball.)

See e.g. Chap. 5 of Nocedal & Wright (2006) and refs therein.

Accelerated First-Order Methods

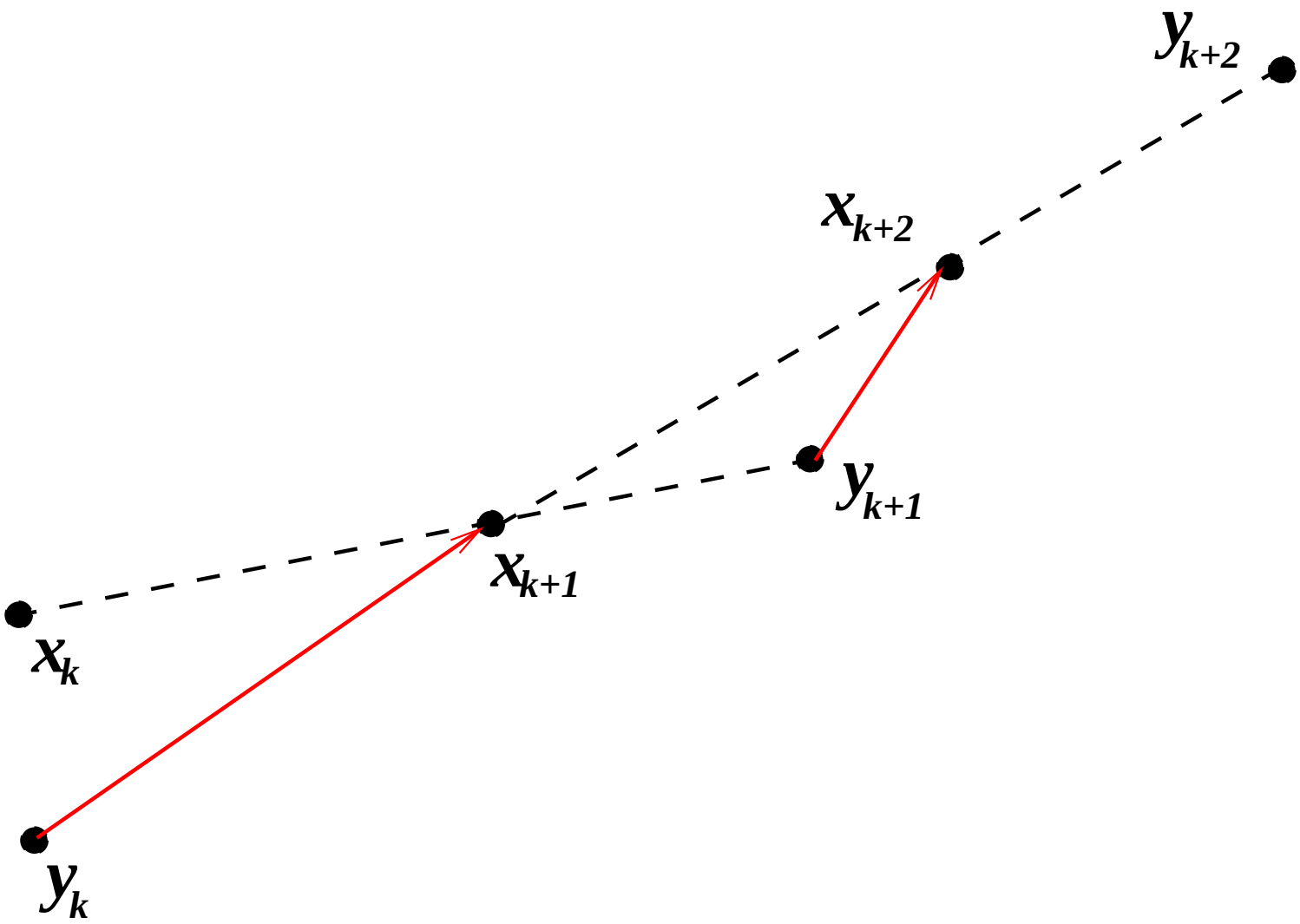
Accelerate the rate to $1/k^2$ for weakly convex, while retaining the linear rate (related to $\sqrt{\kappa}$) for strongly convex case.

Nesterov (1983, 2004) describes a method that requires κ.

0: Choose x_0, $\alpha_0 \in (0,1)$; set $y_0 \leftarrow x_0$./

k: $x_{k+1} \leftarrow y_k - \frac{1}{L}\nabla f(y_k)$; (*short-step gradient*)
solve for $\alpha_{k+1} \in (0,1)$: $\alpha_{k+1}^2 = (1-\alpha_{k+1})\alpha_k^2 + \alpha_{k+1}/\kappa$;
set $\beta_k = \alpha_k(1-\alpha_k)/(\alpha_k^2 + \alpha_{k+1})$;
set $y_{k+1} \leftarrow x_{k+1} + \beta_k(x_{k+1} - x_k)$.

Still works for weakly convex ($\kappa = \infty$).


y_{k+2}
x_{k+2}
y_{k+1}
x_{k+1}
x_k
y_k

Convergence Results: Nesterov

If $\alpha_0 \geq 1/\sqrt{\kappa}$, have

$$f(x_k) - f(x^*) \leq c_1 \min\left(\left(1 - \frac{1}{\sqrt{\kappa}}\right)^k, \frac{4L}{(\sqrt{L} + c_2 k)^2}\right),$$

where constants c_1 and c_2 depend on x_0, α_0, L.

Linear convergence at "heavy-ball" rate in strongly convex case, otherwise $1/k^2$.

In the special case of $\alpha_0 = 1/\sqrt{\kappa}$, this scheme yields

$$\alpha_k \equiv \frac{1}{\sqrt{\kappa}}, \qquad \beta_k \equiv 1 - \frac{2}{\sqrt{\kappa} + 1}.$$

FISTA

(Beck & Teboulle 2007). Similar to the above, but with a fairly short and elementary analysis (though still not very intuitive).

0: Choose x_0; set $y_1 = x_0$, $t_1 = 1$;

k: $x_k \leftarrow y_k - \frac{1}{L}\nabla f(y_k)$;

$t_{k+1} \leftarrow \frac{1}{2}\left(1 + \sqrt{1 + 4t_k^2}\right)$;

$y_{k+1} \leftarrow x_k + \frac{t_k - 1}{t_{k+1}}(x_k - x_{k-1})$.

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 & Teboulle (2010) does the convergence analysis in 2-3 pages: elementary, technical.

A Non-Monotone Gradient Method: Barzilai-Borwein

(Barzilai & Borwein 1988) BB is a gradient method, but with an unusual choice of α_k. Allows f to increase (sometimes dramatically) on some steps.

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

where

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

Explicitly, we have

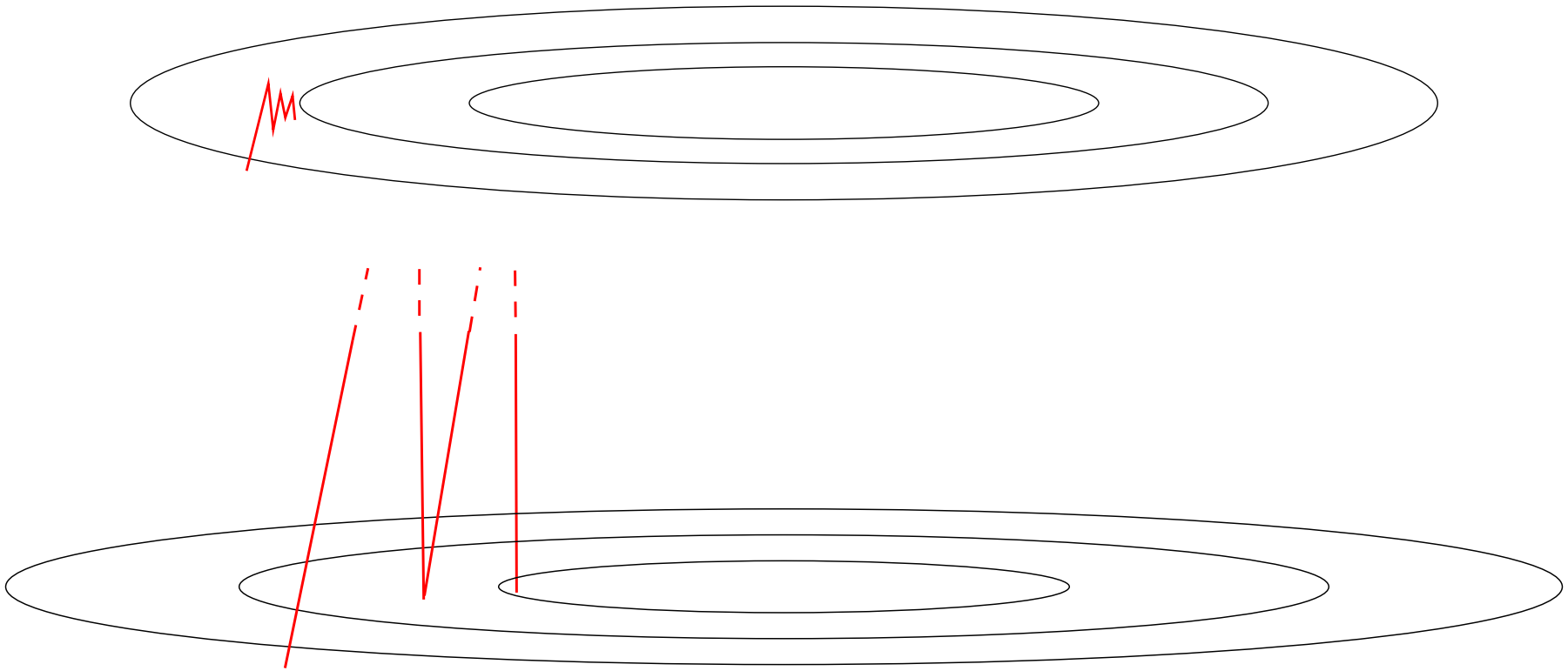
$$\alpha_k = \frac{s_k^T z_k}{z_k^T z_k}.$$

Note that for convex quadratic $f = (1/2)x^T Ax$, we have

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

Hence, can view BB as a kind of quasi-Newton method, with the Hessian approximated by $\alpha_k^{-1} I$.

Comparison: BB vs Greedy Steepest Descent



Many BB Variants

- can 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;
- calculate α_k as above and hold it constant for 2, 3, or 5 successive steps;
- take α_k to be the exact 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 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.

Primal-Dual Averaging

(see Nesterov 2009) Basic step:

$$x_{k+1} = \arg\min_x \frac{1}{k+1}\sum_{i=0}^{k}[f(x_i) + \nabla f(x_i)^T(x - x_i)] + \frac{\gamma}{\sqrt{k}}\|x - x_0\|^2$$
$$= \arg\min_x \bar{g}_k^T x + \frac{\gamma}{\sqrt{k}}\|x - x_0\|^2,$$

where $\bar{g}_k := \sum_{i=0}^{k} \nabla f(x_i)/(k+1)$ — the *averaged gradient*.

- The last term is always centered at the *first* iterate x_0.
- Gradient information is averaged over all steps, with equal weights.
- γ is constant - results can be sensitive to this value.
- The approach still works for convex nondifferentiable f, where $\nabla f(x_i)$ is replaced by a vector from the subgradient $\partial f(x_i)$.

Convergence Properties

Nesterov proves convergence for *averaged* iterates:

$$\bar{x}_{k+1} = \frac{1}{k+1} \sum_{i=0}^{k} x_i.$$

Provided the iterates and the solution x^* lie within some ball of radius D around x_0, we have

$$f(\bar{x}_{k+1}) - f(x^*) \leq \frac{C}{\sqrt{k}},$$

where C depends on D, a uniform bound on $\|\nabla f(x)\|$, and γ (coefficient of stabilizing term).

Note: There's averaging in both primal (x_i) and dual ($\nabla f(x_i)$) spaces.

Generalizes easily and robustly to the case in which only estimated gradients or subgradients are available.

(Averaging smooths the errors in the individual gradient estimates.)

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

How do these methods change when we require $x \in \Omega$, with Ω closed and convex?

Some algorithms and theory stay much the same, provided we can involve Ω explicity in the subproblems.

Example: Primal-Dual Averaging for $\min_{x \in \Omega} f(x)$.

$$x_{k+1} = \arg\min_{x \in \Omega} \bar{g}_k^T x + \frac{\gamma}{\sqrt{k}} \|x - x_0\|^2,$$

where $\bar{g}_k := \sum_{i=0}^{k} \nabla f(x_i)/(k+1)$. When Ω is a box, this subproblem is easy to solve.

Example: Nesterov's Constant Step Scheme for $\min_{x\in\Omega} f(x)$. Requires just only calculation to be changed from the unconstrained version.

0: Choose x_0, $\alpha_0 \in (0,1)$; set $y_0 \leftarrow x_0$, $q \leftarrow 1/\kappa = \mu/L$.

k: $x_{k+1} \leftarrow \arg\min_{y\in\Omega} \frac{1}{2}\|y - [y_k - \frac{1}{L}\nabla f(y_k)]\|_2^2$;
solve for $\alpha_{k+1} \in (0,1)$: $\alpha_{k+1}^2 = (1-\alpha_{k+1})\alpha_k^2 + q\alpha_{k+1}$;
set $\beta_k = \alpha_k(1-\alpha_k)/(\alpha_k^2 + \alpha_{k+1})$;
set $y_{k+1} \leftarrow x_{k+1} + \beta_k(x_{k+1} - x_k)$.

Convergence theory is unchanged.

Regularized Optimization (More Later)

FISTA can be applied with minimal changes to the regularized problem

$$\min_x f(x) + \tau\psi(x),$$

where f is convex and smooth, ψ convex and "simple" but usually nonsmooth, and τ is a positive parameter.

Simply replace the gradient step by

$$x_k = \arg\min_x \frac{L}{2}\left\|x - \left[y_k - \frac{1}{L}\nabla f(y_k)\right]\right\|^2 + \tau\psi(x).$$

(This is the "shrinkage" step; when $\psi \equiv 0$ or $\psi = \|\cdot\|_1$, can be solved cheaply.)

More on this later.

Further Reading

1. Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Kluwer Academic Publishers, 2004.
2. A. Beck and M. Teboulle, "Gradient-based methods with application to signal recovery problems," in press, 2010. (See Teboulle's web site).
3. B. T. Polyak, *Introduction to Optimization*, Optimization Software Inc, 1987.
4. J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," IMA Journal of Numerical Analysis, 8, pp. 141-148, 1988.
5. Y. Nesterov, "Primal-dual subgradient methods for convex programs," Mathematical Programming, Series B, 120, pp. 221-259, 2009.
6. J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed., Springer, 2006.

Stochastic Gradient Methods

Still deal with (weakly or strongly) convex f. But change the rules:

- Allow f nonsmooth.
- Can't get function values $f(x)$.
- At any feasible x, have access only to an 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. Also the special case:

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

where each f_i is convex and nonsmooth.

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

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

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 $\psi(w) = \|w\|_1$.

Subgradients

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

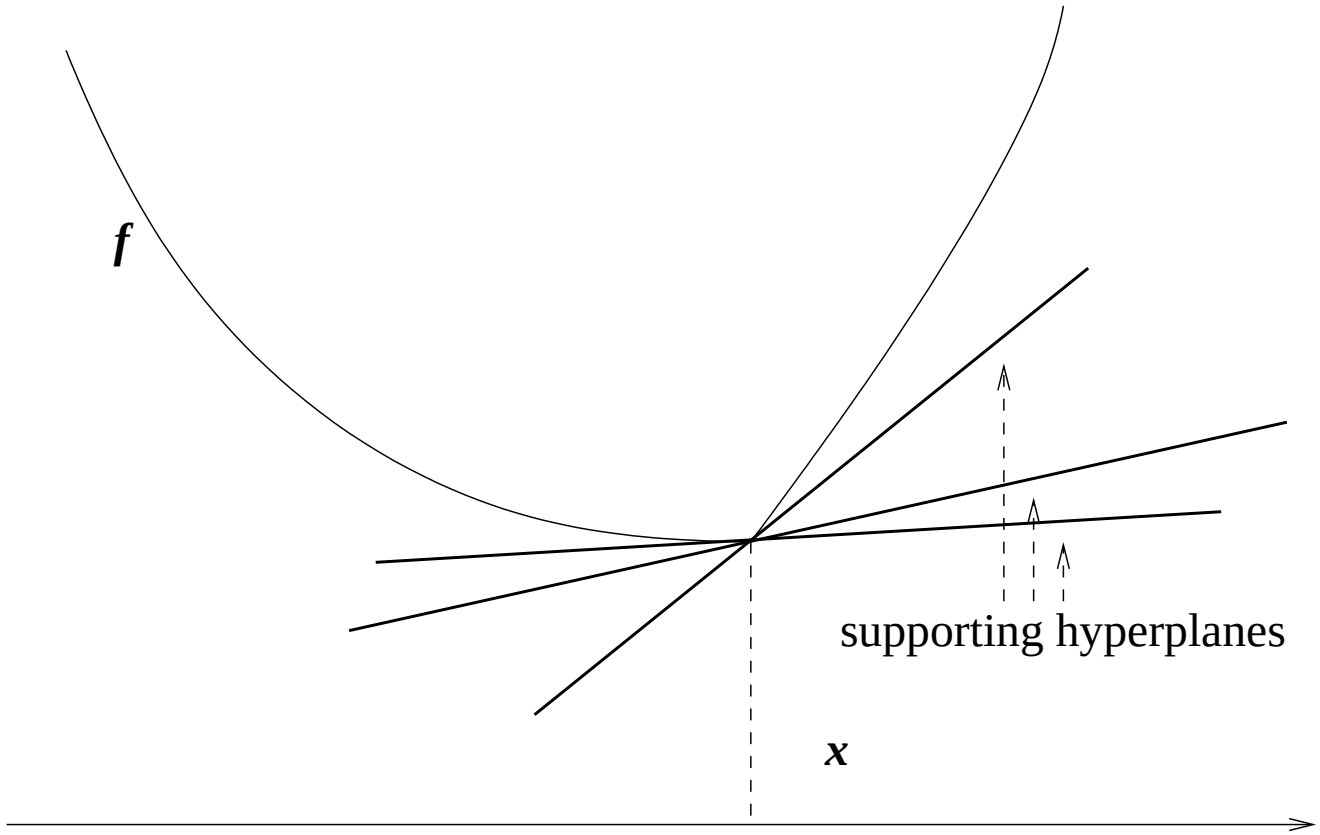
$$f(z) \geq f(x) + g^T(z - x), \qquad \text{for all } z \in \text{dom} f.$$

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

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

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


f
supporting hyperplanes
x

"Classical" Stochastic Approximation

Denote by $G(x, \xi)$ ths subgradient estimate generated at x. For unbiasedness need $E_\xi G(x, \xi) \in \partial f(x)$.

Basic SA Scheme: At iteration k, choose ξ_k i.i.d. according to distribution P, choose some $\alpha_k > 0$, and set

$$x_{k+1} = x_k - \alpha_k G(x_k, \xi_k).$$

Note that x_{k+1} depends on all random variables up to iteration k, i.e. $\xi_{[k]} := \{\xi_1, \xi_2, \ldots, \xi_k\}$.

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

Rate: $1/k$

Define $a_k = \frac{1}{2}E(\|x_k - x^*\|^2)$. Assume there is $M > 0$ such that $E(\|G(x,\xi)\|^2) \leq M^2$ for all x of interest. Thus

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

Taking expectations, get

$$a_{k+1} \leq a_k - \alpha_k E[(x_k - x^*)^T G(x_k,\xi_k)] + \frac{1}{2}\alpha_k^2 M^2.$$

For middle term, have

$$\begin{aligned}
E[(x_k - x^*)^T G(x_k,\xi_k)] &= E_{\xi_{[k-1]}} E_{\xi_k}[(x_k - x^*)^T G(x_k,\xi_k)|\xi_{[k-1]}] \\
&= E_{\xi_{[k-1]}} (x_k - x^*)^T g_k,
\end{aligned}$$

... where

$$g_k := E_{\xi_k}[G(x_k, \xi_k)|\xi_{[k-1]}] \in \partial f(x_k).$$

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:

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

Proof: Clearly true for $k = 1$. Otherwise:

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

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*.

Robust SA

At iteration k:

- set $x_{k+1} = x_k - \alpha_k G(x_k, \xi_k)$ as before;
- set

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

For any $\theta > 0$ (not critical), 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:

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

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

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

thus

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

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

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

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

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

By convexity, we have

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

so obtain the fundamental bound:

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

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

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

That's it!

Other variants: constant stepsizes α_k for a fixed "budget" of iterations; periodic restarting; averaging just over the recent iterates. All 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 and without requiring an accurate estimate of μ. The tricks are to (a) define the desired threshold for a_k in advance and (b) use a constant step size

Recall the bound from a few slides back, and set $\alpha_k \equiv \alpha$:

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

Define the "limiting value" α_∞ by

$$a_\infty = (1 - 2\mu\alpha)a_\infty + \frac{1}{2}\alpha^2 M^2.$$

Take the difference of the two expressions above:

$$(a_{k+1} - a_\infty) \leq (1 - 2\mu\alpha)(a_k - a_\infty)$$

from which it follows that $\{a_k\}$ decreases monotonically to a_∞, and

$$(a_k - a_\infty) \leq (1 - 2\mu\alpha)^k (a_0 - a_\infty).$$

Constant Step Size, continued

Rearrange the expression for a_∞ to obtain

$$a_\infty = \frac{\alpha M^2}{4\mu}.$$

From the previous slide, we thus have

$$\begin{aligned} a_k &\leq (1 - 2\mu\alpha)^k (a_0 - a_\infty) + a_\infty \\ &\leq (1 - 2\mu\alpha)^k a_0 + \frac{\alpha M^2}{4\mu}. \end{aligned}$$

Given threshold $\epsilon > 0$, we aim to find α and K such that $a_k \leq \epsilon$ for all $k \geq K$. We ensure that both terms on the right-hand side of the expression above 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:

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

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.)

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.*

Mirror Descent

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

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

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. In some cases (e.g. when Ω is a box), the subproblem is still easy to solve.

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.

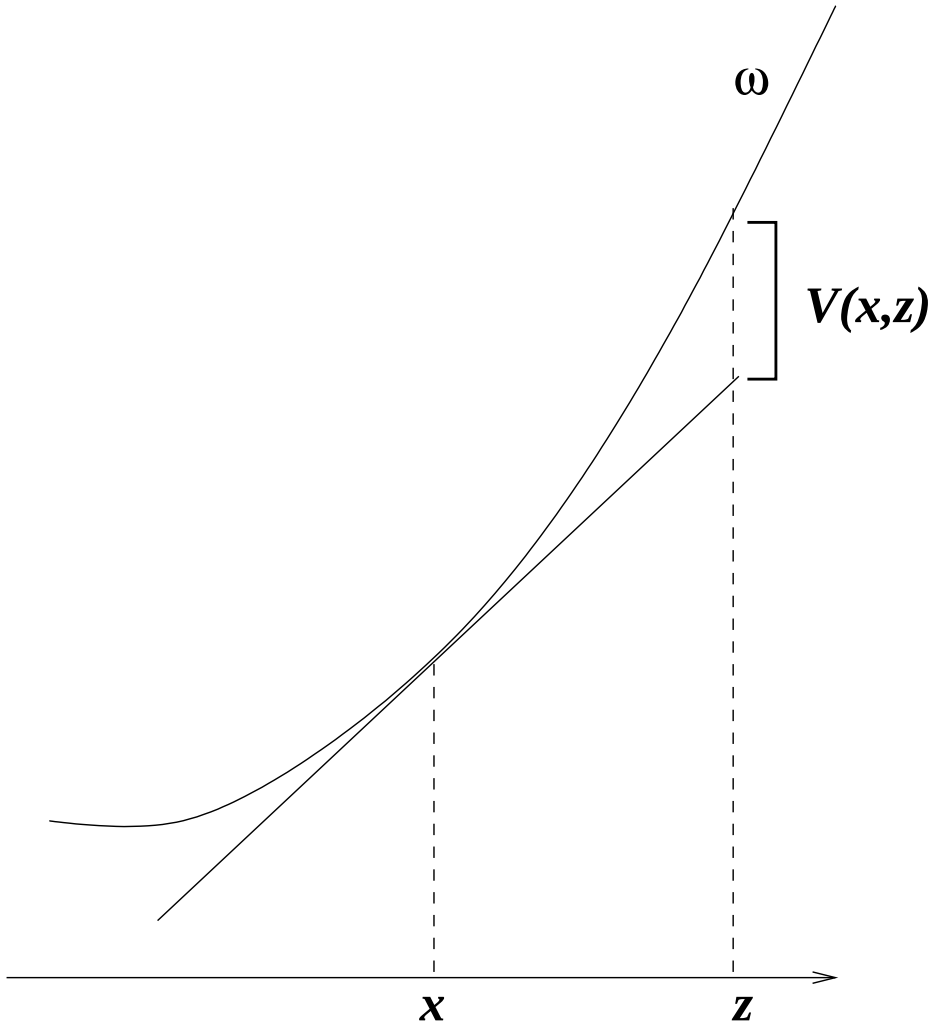
Now define the *prox-function* $V(x, z)$ as follows:

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

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

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

Bregman distance is the deviation from linearity:



Bregman Distances: Examples

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

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

These are the two most useful cases.

Convergence results for SA can be generalized to mirror descent.

Incremental Gradient

(See e.g. Bertsekas (2011) and references therein.) Finite sums:

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

Step k typically requires choice of one index $i_k \in \{1, 2, \ldots, m\}$ and evaluation of $\nabla f_{i_k}(x_k)$. Components i_k are selected sometimes randomly or cyclically. (Latter option does not exist in the setting $f(x) := E_\xi F(x; \xi)$.)

- There are incremental versions of the heavy-ball method:

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

- Approach like dual averaging: assume a cyclic choice of i_k, and approximate $\nabla f(x_k)$ by the average of $\nabla f_i(x)$ over the last m iterates:

$$x_{k+1} = x_k - \frac{\alpha_k}{m} \sum_{l=1}^{m} \nabla f_{i_{k-l+1}}(x_{k-l+1}).$$

Achievable Accuracy

Consider the basic incremental method:

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

How close can $f(x_k)$ come to $f(x^*)$ — deterministically (not just in expectation).

Bertsekas (2011) obtains results for constant steps $\alpha_k \equiv \alpha$.

$$\text{cyclic choice of } i_k: \quad \liminf_{k\to\infty} f(x_k) \leq f(x^*) + \alpha\beta m^2 c^2.$$

$$\text{random choice of } i_k: \quad \liminf_{k\to\infty} f(x_k) \leq f(x^*) + \alpha\beta m c^2.$$

where β is close to 1 and c is a bound on the Lipschitz constants for ∇f_i.

(Bertsekas actually proves these results in the more general context of regularized optimization - see below.)

Applications to SVM

SA techniques have an obvious application to linear SVM classification. In fact, they were proposed in this context and analyzed independently by researchers in the ML community for some years.

Codes: SGD (Bottou), PEGASOS (Shalev-Schwartz et al, 2007).

Tutorial: *Stochastic Optimization for Machine Learning*, Tutorial by N. Srebro and A. Tewari, ICML 2010 for many more details on the connections between stochastic optimization and machine learning.

Related Work: Zinkevich (ICML, 2003) on online convex programming. Aiming to approximate the minimize the average of a sequence of convex functions, presented sequentially. No i.i.d. assumption, regret-based analysis. Take steplengths of size $O(k^{-1/2})$ in gradient $\nabla f_k(x_k)$ of latest convex function. Average regret is $O(k^{-1/2})$.

Parallel Stochastic Approximation

Several approaches tried for parallel stochastic approximation.

- **Dual Averaging:** Average gradient estimates evaluated in parallel on different cores. Requires message passing / synchronization (Dekel et al, 2011; Duchi et al, 2010).
- **Round-Robin:** Cores evaluate ∇f_i in parallel and update centrally stored x in round-robin fashion. Requires synchronization (Langford et al, 2009).
- **Asynchronous:** HOGWILD!**:** Each core grabs the centrally-stored x and evaluates $\nabla f_e(x_e)$ for some random e, then writes the updates back into x (Niu, Ré, Recht, Wright, NIPS, 2011).

HOGWILD!: Each processor runs independently:

1. Sample e from E;
2. Read current state of x;
3. **for** v in e **do** $x_v \leftarrow x_v - \alpha[\nabla f_e(x_e)]_v$;

HOGWILD! Convergence

- Updates can be old by the time they are applied, but we assume a bound τ on their age.
- Nui et al (2011) analyze the case in which the update is applied to just one $v \in e$, but can be extended easily to update the full edge e, provided this is done atomically.
- Processors can overwrite each other's work, but sparsity of ∇f_e helps — updates to not interfere too much.

Analysis of Niu et al (2011) recently simplified and generalized by Richtarik (2012).

In addition to L, μ, M, D_0 defined above, also define quantities that capture the size and interconnectivity of the subvectors x_e.

- $\rho_e = |\{e' : e' \cap e \neq \emptyset\}|$: number of indices e' such that x_e and $x_{e'}$ have common components;
- $\rho = \sum_{e \in E} \rho_e / |E|^2$: average rate of overlapping subvectors.

HOGWILD! Convergence

(Richtarik 2012) (for full atomic update of index e) Given $\epsilon \in (0, D_0/L)$, we have

$$\min_{0 \le j \le k} E(f(x_j) - f(x^*)) \le \epsilon,$$

for

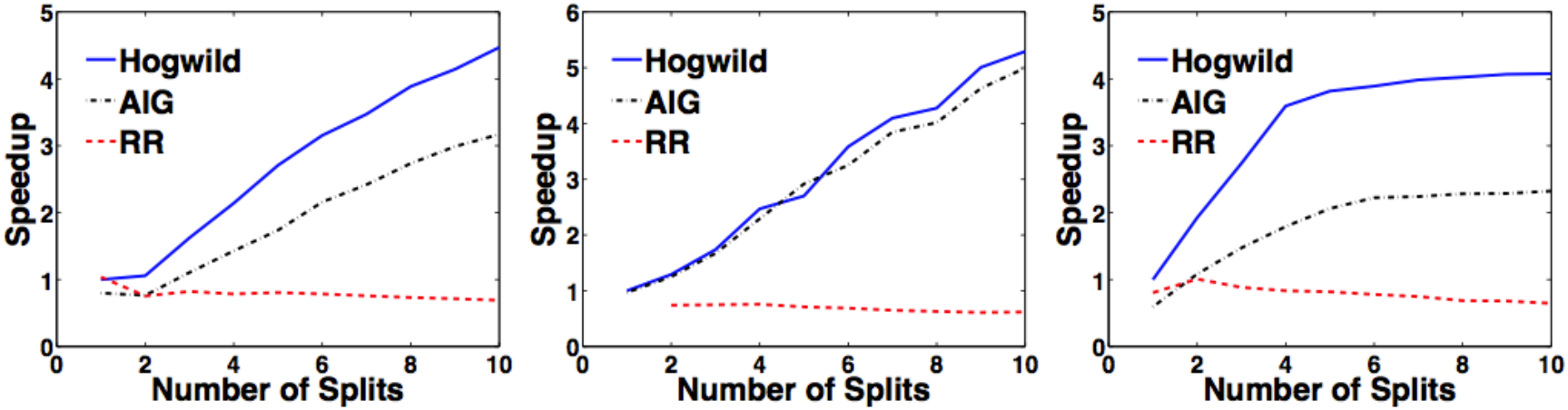
$$\alpha_k \equiv \frac{\mu\epsilon}{(1 + 2\tau\rho)LM^2|E|^2}$$

and $k \ge K$, where

$$K = \frac{(1 + 2\tau\rho)LM^2|E|^2}{\mu^2\epsilon} \log\left(\frac{2LD_0}{\epsilon} - 1\right).$$

Broadly, recovers the sublinear $1/k$ convergence rate seen in regular SGD, with the delay τ and overlap measure ρ both appearing linearly.

HOGWILD! Performance



HOGWILD! compared with averaged gradient (AIG) and round-robin (RR). Experiments run on a 12-core machine. (10 cores used for gradient evaluations, 2 cores for data shuffling.)

HOGWILD! Performance

	data set	size (GB)	ρ	Δ	time (s)	speedup
SVM	RCV1	0.9	4.4E-01	1.0E+00	10	4.5
	Netflix	1.5	2.5E-03	2.3E-03	301	5.3
MC	KDD	3.9	3.0E-03	1.8E-03	878	5.2
	JUMBO	30	2.6E-07	1.4E-07	9,454	6.8
	DBLife	0.003	8.6E-03	4.3E-03	230	8.8
CUTS	Abdomen	18	9.2E-04	9.2E-04	1,181	4.1

Extensions

To improve scalability, could restrict write access.

- Break x into blocks; assign one block per processor; allow a processor to update only components in its block;
- Share blocks by periodically writing to a central repository, or gossipping between processors.

Analysis in progress.

Le et al (2012) (featured recently in the NY Times) implemented an algorithm like this on 16,000 cores.

Another useful tool for splitting problems and coordinating information between processors is the Alternating Direction Method of Mulitipliers (ADMM).

Further Reading

1. A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," SIAM Journal on Optimization, 19, pp. 1574-1609, 2009.
2. D. P. Bertsekas, "Incremental gradient, subgradient, and proximal methods for convex optimization: A Survey," Chapter 4 in *Optimization and Machine Learning,* S. Nowozin, S. Sra, and S. J. Wright (2011).
3. A. Juditsky and A. Nemirovski, " First-order methods for nonsmooth convex large-scale optimization. I and II" methods," Chapters 5 and 6 in *Optimization and Machine Learning* (2011).
4. O. L. Mangasarian and M. Solodov, "Serial and parallel backpropagation convergencevia nonmonotone perturbed minimization," Optimization Methods and Software 4 (1994), pp. 103–116.
5. D. Blatt, A. O. Hero, and H. Gauchman, "A convergent incremental gradient method with constant step size," SIAM Journal on Optimization 18 (2008), pp. 29–51.
6. Niu, F., Recht, B., Ré, C., and Wright, S. J., "HOGWILD!: A Lock-free approach to parallelizing stochastic gradient descent," NIPS 24, 2011.