Optimization Algorithms in Machine Learning

Stephen Wright

University of Wisconsin-Madison

NIPS Tutorial, 6 Dec 2010

Optimization

- Optimization is going through a period of growth and revitalization, driven largely by new applications in many areas.
- Standard paradigms (LP, QP, NLP, MIP) are still important, along with general-purpose software, enabled by modeling languages that make the software easier to use.
- However, there is a growing emphasis on "picking and choosing" algorithmic elements to fit the characteristics of a given application — building up a suitable algorithm from a "toolkit" of components.
- It's more important than ever to understand the fundamentals of algorithms as well as the demands of the application, so that good choices are made in matching algorithms to applications.

We present a selection of algorithmic fundamentals in this tutorial, with an emphasis on those of current and potential interest in machine learning.

Topics

I. First-order Methods
II. Stochastic and Incremental Gradient Methods
III. Shrinking/Thresholding for Regularized Formulations
IV. Optimal Manifold Identification and Higher-Order Methods.
V. Decomposition and Coordinate Relaxation

I. First-Order 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.

Gradient

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

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.

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

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

It follows by elementary arguments (see e.g. Nesterov 2004) that

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

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

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

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 $k+2$ entries. Hence, $x_{k+1}(i) = 0$ for $i = k+3, k+4, \ldots, n$. 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 $\left[\frac{1}{L}, \frac{1}{\mu}\right]$. Can show that

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

No improvement in the linear rate over constant steplength.

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

Best 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.

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

There is a rich convergence theory for f quadratic, including 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 $y_0 \leftarrow x_0$./

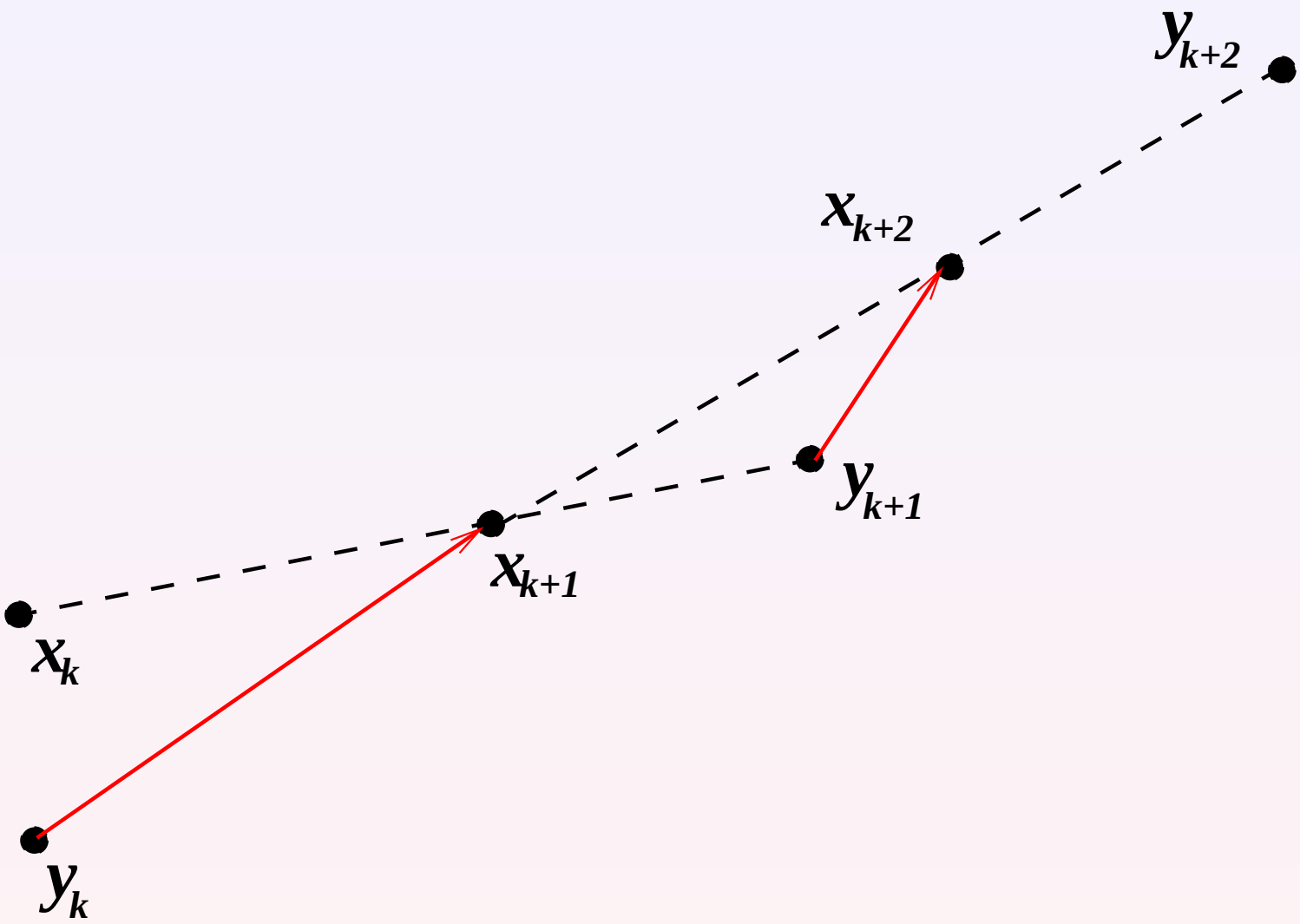
k: $x_{k+1} \leftarrow y_k - \frac{1}{L}\nabla f(y_k)$; (*short-step gradient*)
solve for $\alpha_{k+1} \in (0,1)$: $\alpha_{k+1}^2 = (1-\alpha_{k+1})\alpha_k^2 + \alpha_{k+1}/\kappa$;
set $\beta_k = \alpha_k(1-\alpha_k)/(\alpha_k^2 + \alpha_{k+1})$;
set $y_{k+1} \leftarrow x_{k+1} + \beta_k(x_{k+1}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

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

FISTA (Beck & Teboulle 2007):

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

k: $x_k \leftarrow y_k - \frac{1}{L}\nabla f(y_k)$;
$t_{k+1} \leftarrow \frac{1}{2}\left(1 + \sqrt{1+4t_k^2}\right)$;
$y_{k+1} \leftarrow x_k + \frac{t_k - 1}{t_{k+1}}(x_k - x_{k-1})$.


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 $1/k^2$. FISTA also achieves $1/k^2$ rate.

Analysis: Not intuitive. Based on bounding the difference between f and a quadratic approximation to it, at x^*. FISTA analysis is 2-3 pages.

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

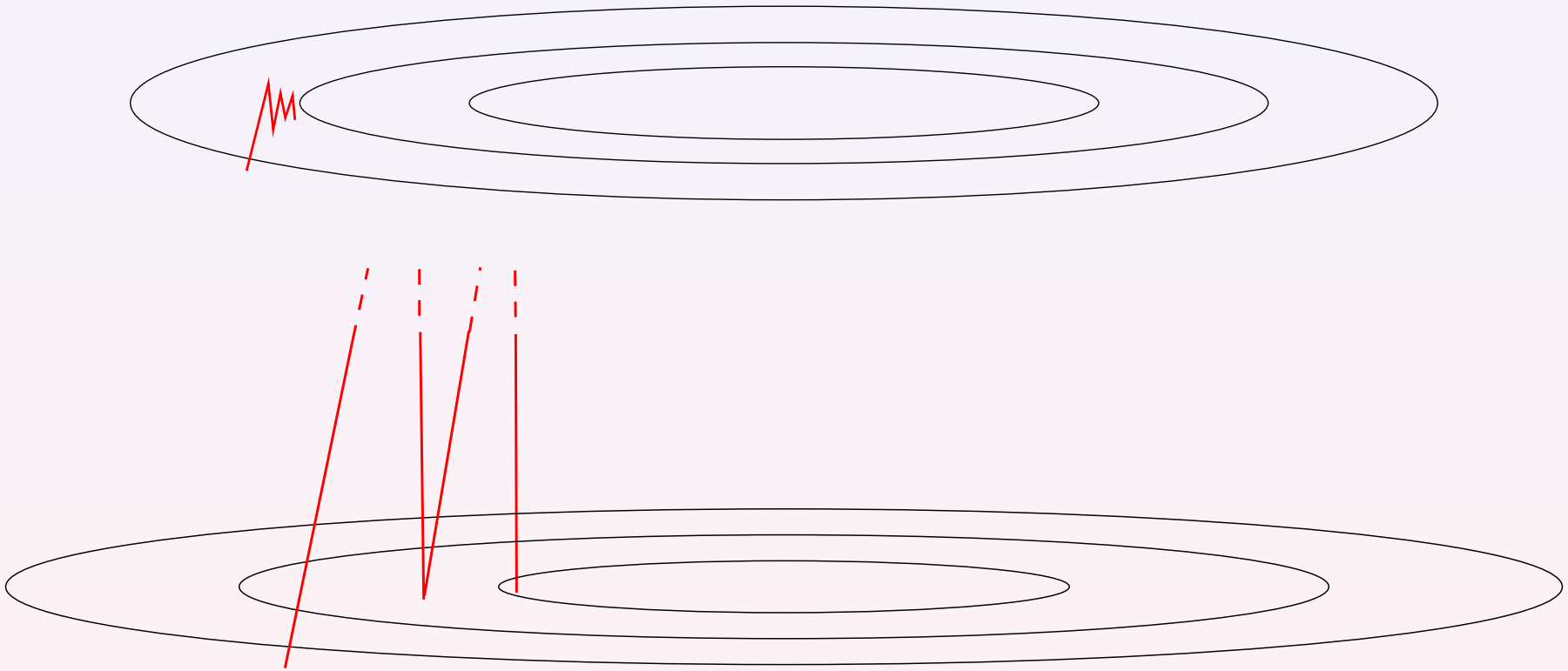
$$\alpha_k = \frac{s_k^T z_k}{z_k^T z_k}.$$

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.

II. Stochastic and Incremental 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

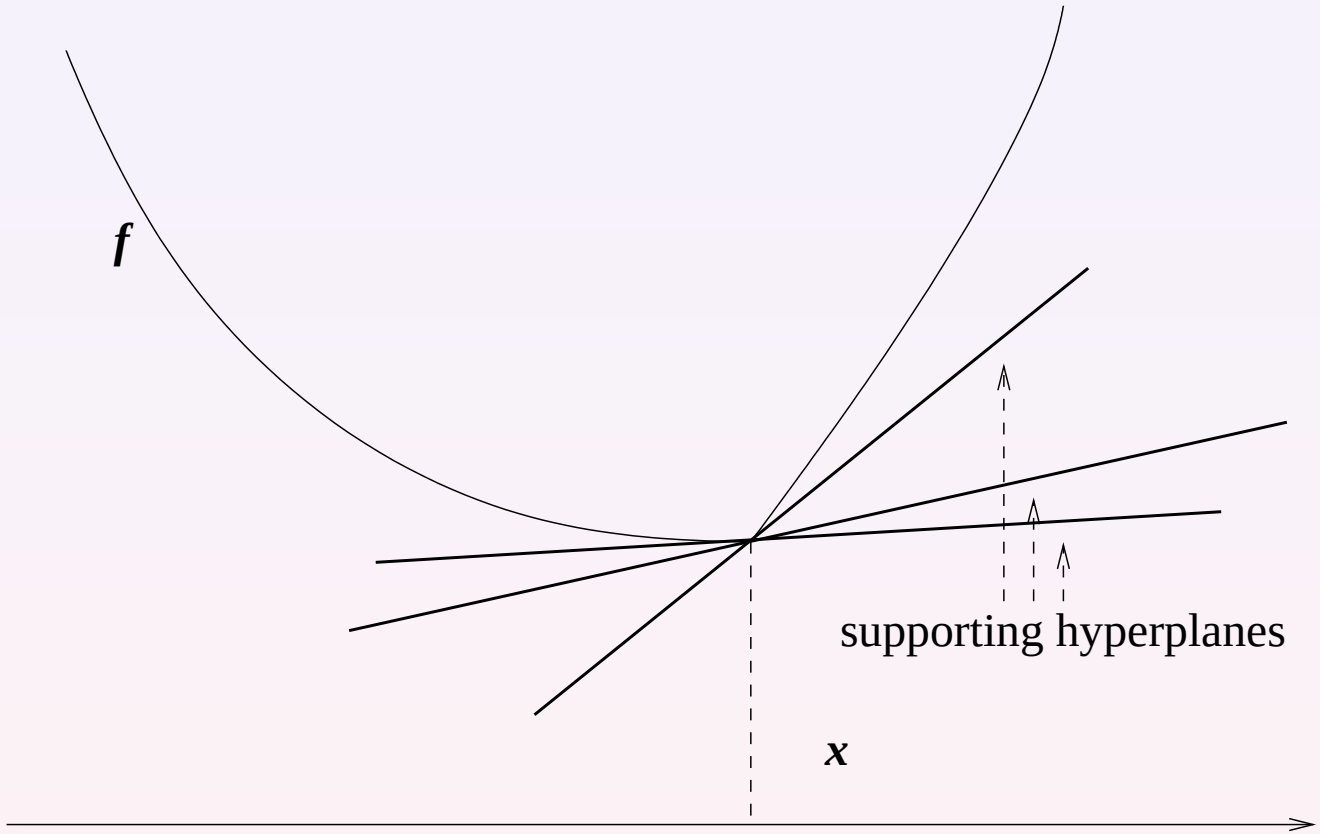
$$f(z) \geq f(x) + g^T(z - x), \qquad \text{for all } z \in \text{dom} f.$$

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 (exercise!) reveals the $1/k$ rate:

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

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.

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.

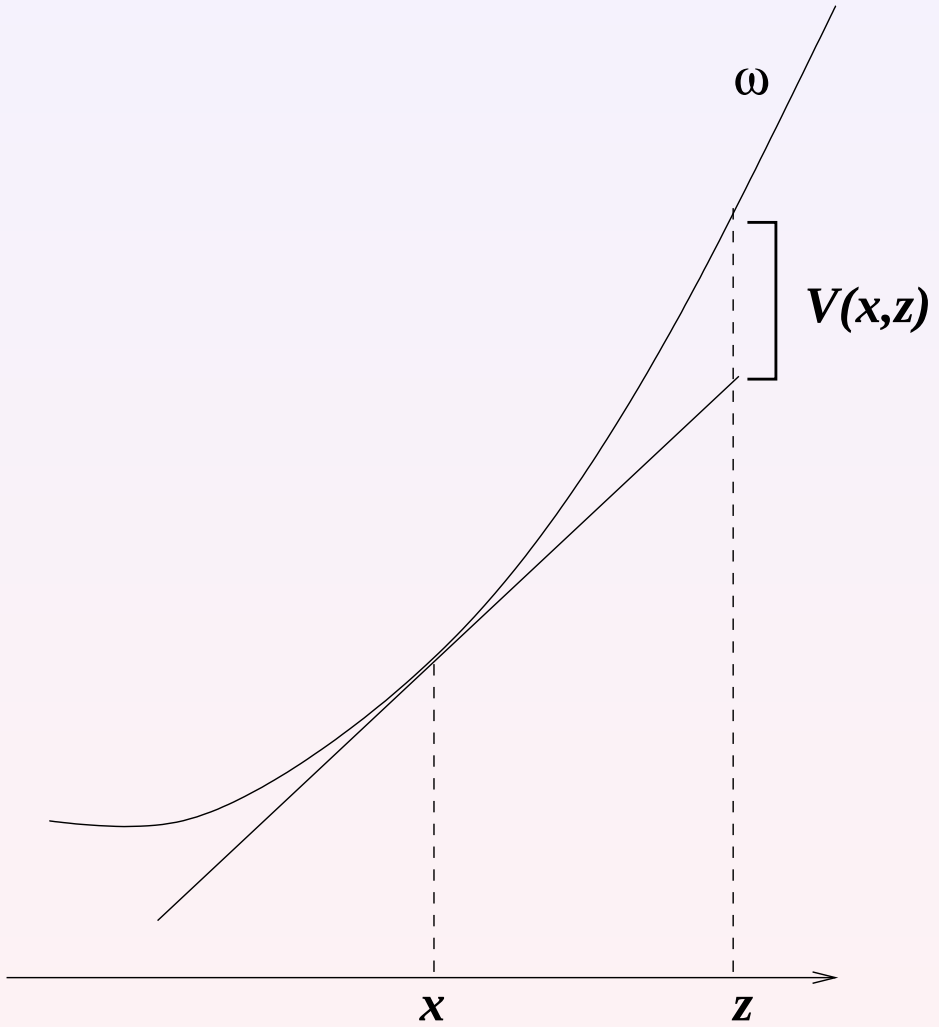
Now define the *prox-function* $V(x, z)$ as follows:

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

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 $O(k^{-1/2})$.

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,* upcoming volume edited by 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: General-purpose methods," Chapter 5 in *Optimization and Machine Learning* (2011).
4. A. Juditsky and A. Nemirovski, " First-order methods for nonsmooth convex large-scale optimization. I: Utilizing problem structure," Chapter 6 in *Optimization and Machine Learning* (2011).
5. O. L. Mangasarian and M. Solodov, "Serial and parallel backpropagation convergencevia nonmonotone perturbed minimization," Optimization Methods and Software 4 (1994), pp. 103–116.
6. 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.

III. Shrinking/Thresholding for Regularized Optimization

In many applications, we seek not an exact minimizer of the underlying objective, but rather an approximate minimizer that satisfies certain desirable properties:

- sparsity (few nonzeros);
- low-rank (if a matrix);
- low "total-variation";
- generalizability. (Vapnik: "...tradeoff between the quality of the approximation of the given data and the complexity of the approximating function.")

"Desirable" properties depend on context and application .

A common way to obtain structured solutions is to modify the objective f by adding a regularizer $\tau\psi(x)$, for some parameter $\tau > 0$.

$$\min f(x) + \tau\psi(x).$$

Often want to solve for a range of τ values, not just one value in isolation.

Basics of Shrinking

Regularizer ψ is often nonsmooth but "simple." Shrinking / thresholding approach (a.k.a. forward-backward splitting) is useful if the problem is easy to solve when f is replaced by a quadratic with diagonal Hessian:

$$\min_z \; g^T(z-x) + \frac{1}{2\alpha}\|z-x\|_2^2 + \tau\psi(z).$$

Equivalently,

$$\min_z \; \frac{1}{2\alpha}\|z-(x-\alpha g)\|_2^2 + \tau\psi(z).$$

Define the shrinking operator as the arg min:

$$S_\tau(y,\alpha) := \arg\min_z \; \frac{1}{2\alpha}\|z-y\|_2^2 + \tau\psi(z).$$

Typical algorithm:

$$x_{k+1} = S_\tau(x_k - \alpha_k g_k, \alpha_k),$$

with for example $g_k = \nabla f(x_k)$.

"Practical" Instances of ψ

Cases for which the subproblem is simple:

- $\psi(z) = \|z\|_1$. Thus $S_\tau(y, \alpha) = \text{sign}(y)\max(|y| - \alpha\tau, 0)$. When y complex, have

$$S_\tau(y, \alpha) = \frac{\max(|y| - \tau\alpha, 0)}{\max(|y| - \tau\alpha, 0) + \tau\alpha} y.$$

- $\psi(z) = \sum_{g \in G} \|z_{[g]}\|_2$ or $\psi(z) = \sum_{g \in G} \|z_{[g]}\|_\infty$, where $z_{[g]}$, $g \in G$ are non-overlapping subvectors of z. Here

$$S_\tau(y, \alpha)_{[g]} = \frac{\max(|y_{[g]}| - \tau\alpha, 0)}{\max(|y_{[g]}| - \tau\alpha, 0) + \tau\alpha} y_{[g]}.$$

- Z is a matrix and $\psi(Z) = \|Z\|_*$ is the nuclear norm of Z: the sum of singular values. Threshold operator is

$$S_\tau(Y, \alpha) := \arg\min_Z \frac{1}{2\alpha}\|Z - Y\|_F^2 + \tau\|Z\|_*$$

with solution obtained from the SVD $Y = U\Sigma V^T$ with U, V orthonormal and $\Sigma = \text{diag}(\sigma_i)_{i=1,2,\ldots,m}$. Setting $\tilde{\Sigma} = \text{diag}(\max(\sigma_i - \tau\alpha, 0)_{i=1,2,\ldots,m})$, the solution is

$$S_\tau(Y, \alpha) = U\tilde{\Sigma}V^T.$$

(Actually not cheap to compute, but in some cases (e.g. $\sigma_i - \tau\alpha < 0$ for most i) approximate solutions can be found in reasoable time.)

Connections

The thresholding operator generalizes:

- Gradient methods for unconstrained minimization. Here $\psi \equiv 0$ and $S_\tau(y, \alpha) = y$.
- Projected gradient for $\min_{x \in \Omega} f(x)$ with Ω closed and convex. Here ψ is the indicator function for Ω (zero on Ω, ∞ elsewhere), and

$$S_\tau(y, \alpha) = P_\Omega(y),$$

where P_Ω is projection onto Ω.

Applications

LASSO for variable selection. Originally stated as

$$\min_x \frac{1}{2}\|Ax - b\|_2^2 \text{ such that } \|x\|_1 \leq T,$$

for parameter $T > 0$. Equivalent to an "ℓ_2-ℓ_1" formulation:

$$\min_x \frac{1}{2}\|Ax - b\|_2^2 + \tau\|x\|_1, \qquad \text{for some } \tau > 0.$$

Group LASSO for selection of variable "groups."

$$\min_x \frac{1}{2}\|Ax - b\|_2^2 + \sum_{g \in G} \|x_{[g]}\|_2,$$

with each $[g]$ a subset of indices $\{1, 2, \ldots, n\}$.

- When groups $[g]$ are disjoint, easy to solve the subproblem.
- Still true if $\|\cdot\|_2$ is replaced by $\|\cdot\|_\infty$.
- When groups *overlap*, can replicate variables, to have one copy of each variable in each group — thus reformulate as non-overlapping.

Compressed Sensing. Sparse signal recovery from noisy measurements. Given matrix A (with more columns than rows) and observation vector y, seek a sparse x (i.e. few nonzeros) such that $Ax \approx y$. Solve

$$\min_x \frac{1}{2}\|Ax - b\|_2^2 + \tau\|x\|_1.$$

- Under "restricted isometry" properties on A ("tall, thin" column submatrices are nearly orthonormal), $\|x\|_1$ is a good surrogate for $\text{card}(x)$.
- Assume that A is not stored explicitly, but matrix-vector multiplications are available. Hence can compute f and ∇f.
- Often need solution for a range of τ values.

ℓ_1-Regularized Logistic Regression. Feature vectors x_i, $i = 1, 2, \ldots, m$ with labels ± 1. Seek odds function parametrized by $w \in \mathbb{R}^n$:

$$p_+(x; w) := (1 + e^{w^T x})^{-1}, \quad p_-(x; w) := 1 - p_+(x; w).$$

Scaled, negative log likelihood function $\mathcal{L}(w)$ is

$$\begin{aligned}\mathcal{L}(w) &= -\frac{1}{m}\left[\sum_{y_i=-1} \log p_-(x_i; w) + \sum_{y_i=1} \log p_+(x_i; w)\right] \\ &= -\frac{1}{m}\left[\sum_{y_i=-1} w^T x_i - \sum_{i=1}^{m} \log(1 + e^{w^T x_i})\right].\end{aligned}$$

To get a sparse w (i.e. classify on the basis of a few features) solve:

$$\min_w \mathcal{L}(w) + \lambda \|w\|_1.$$

Matrix Completion. Seek a matrix $X \in \mathbb{R}^{m \times n}$ with low rank that matches certain observations, possibly noisy.

$$\min_X \frac{1}{2}\|\mathcal{A}(X) - b\|_2^2 + \tau\psi(X),$$

where $\mathcal{A}(X)$ is a linear mapping of the components of X (e.g. element-wise observations).

Can have ψ as the nuclear norm — see discussion above for solution of subproblems via SVD.

At NIPS 2010: "Practical Large-Scale Optimization for Max-Norm Regularization" by Lee et al. discuss $\psi(X) = \|X\|_{\max}$:

$$\|X\|_{\max} := \inf\{\|U\|_{2,\infty}, \|V\|_{2,\infty} \mid X = UV^T\},$$

where $\|U\|_{2,\infty}$ is the maximum ℓ_2 norm of a row of U. The shrinking operation can be solved efficiently using the "squash" operator.

Basic Algorithm

(Fukushima and Mine, 1981) for solving $\min_x f(x) + \tau\psi(x)$.

0: Choose x_0

k: Choose $\alpha_k > 0$ and set

$$\begin{aligned} x_{k+1} &= S_\tau(x_k - \alpha_k \nabla f(x_k); \alpha_k) \\ &= \arg\min_z \nabla f(x_k)^T(z - x_k) + \frac{1}{2\alpha_k}\|z - x_k\|_2^2 + \tau\psi(z). \end{aligned}$$

Straightforward, but can be fast when the regularization is strong (i.e. solution is "highly constrained").

Can show convergence for steps $\alpha_k \in (0, 2/L)$, where L is the bound on $\nabla^2 f$. (Like a short-step gradient method.)

Enhancements

Alternatively, since α_k plays the role of a steplength, can adjust it to get better performance *and* guaranteed convergence.

- "Backtracking:" decrease α_k until sufficient decrease condition holds.
- Use Barzilai-Borwein strategies to get nonmonotonic methods. By enforcing sufficient decrease every 10 iterations (say), still get global convergence.

The approach can be accelerated using optimal gradient techniques. See earlier discussion of **FISTA**, where we solve the shrinking problem with $\alpha_k = 1/L$ in place of a step along $-\nabla f$ with this steplength.

Note that these methods reduce ultimately to gradient methods on a reduced space: the optimal manifold defined by the regularizer ψ. Acceleration or higher-order information can help improve performance.

Continuation in τ

Performance of basic shrinking methods is quite sensitive to τ.

Typically higher $\tau \Rightarrow$ stronger regularization $\Rightarrow$ optimal manifold has lower dimension. Hence, it's easier to identify the optimal manifold, and basic shrinking methods can sometimes do so quickly.

For smaller τ, a simple "continuation" strategy can help:

0: Given target value τ_f, choose initial $\tau_0 > \tau_f$, starting point $\bar{x}$ and factor $\sigma \in (0, 1)$.

k: Find approx solution $x(\tau_k)$ of $\min_x f(x) + \tau\psi(x)$, starting from $\bar{x}$;
if $\tau_k = \tau_f$ **then STOP**;
Set $\tau_{k+1} \leftarrow \max(\tau_f, \sigma\tau_k)$ and $\bar{x} \leftarrow x(\tau_k)$;

- Solution $x(\tau)$ is often desired on a range of τ values anyway, so efforts for larger τ are not wasted.
- Accelerated methods such as FISTA are less sensitive to the "small τ" issue.
- Not much analysis of this approach has been done. Better heuristics and theoretical support are needed.

Stochastic Gradient + Regularization

Solve the regularized problem, but have only *estimates* of $\nabla f(x_k)$.

We can combine dual averaging, stochastic gradient, and shrinking: see Xiao (2010).

$$\min_x \phi_\tau(x) := E_\xi f(x;\xi) + \tau\psi(x)$$

At iteration k choose ξ_k randomly and i.i.d from the ξ distribution, and choose $g_k \in \partial f(x_k;\xi_k)$. Use these to define the averaged subgradient $\bar{g}_k = \sum_{i=1}^k g_i/(k+1)$, and solve the subproblem

$$x_{k+1} = \arg\min_x \bar{g}_k^T x + \tau\psi(x) + \frac{\gamma}{\sqrt{k}}\|x - x_0\|^2.$$

Same as earlier, but with regularizer ψ included explicitly.

Can prove convergence results for averaged iterates $\bar{x}_k$: roughly

$$E\phi_\tau(\bar{x}_k) - \phi_\tau^* \le \frac{C}{\sqrt{k}},$$

where the expectation of ϕ is taken over the random number stream $\xi_0, \xi_1, \ldots, \xi_{k-1}$.

Further Reading

1. F. Bach. "Sparse methods for machine learning: Theory and algorithms" Tutorial at NIPS 2009. (Slides on web.)
2. M. Fukushima and H. Mine. "A generalized proximal point algorithm for certain non-convex minimization problems." International Journal of Systems Science, 12, pp. 989–1000, 1981.
3. P. L. Combettes and V. R. Wajs. "Signal recovery by proximal forward-backward splitting." Multiscale Modeling and Simulation, 4, pp. 1168–1200, 2005.
4. S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo. "Sparse reconstruction by separable approximation." IEEE Transactions on Signal Processing, 57, pp. 2479–2493, 2009.
5. E. Candès, J. Romberg, and T. Tao. "Stable signal recovery for incomplete and inaccurate measurements." Communications in Pure and Applied Mathematics, 59, pp. 1207–1223, 2006.
6. L. Xiao. "Dual averaging methods for regularized stochastic learning and online optimization." TechReport MSR-TR-2010-23, Microsoft Research, March 2010.

IV. Optimal Manifold Identification

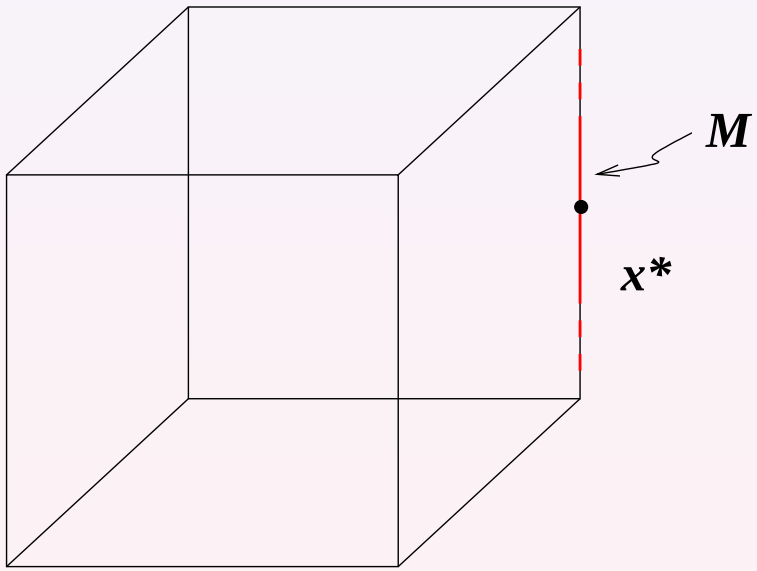
When constraints $x \in \Psi$ or a nonsmooth regularizer $\psi(x)$ are present, identification of the manifold on which x^* lies can improve algorithm performance, by focusing attention on a reduced space. We can thus evaluate *partial* gradients and Hessians, restricted to just this space.

For nonsmooth regularizer ψ, the active manifold is a smooth surface passing through x^* along which ψ is smooth.
Example: for $\psi(x) = \|x\|_1$, have manifold consisting of z with

$$z_i \begin{cases} \geq 0 & \text{if } x_i^* > 0 \\ \leq 0 & \text{if } x_i^* < 0 \\ = 0 & \text{if } x_i^* = 0. \end{cases}$$

For a polyhedral Ω, the active manifold is the face on which x^* lies.
Example: For $\Omega = [0,1]^n$, active manifold consists of z with

$$z_i \begin{cases} = 1 & \text{if } x_i^* = 1 \\ = 0 & \text{if } x_i^* = 0 \\ \in [0,1] & \text{if } x_i^* \in (0,1). \end{cases}$$



Can parametrize $\mathcal{M}$ with a single variable.

Identification

Algorithms of shrinking / gradient projection type can identify the optimal manifold $\mathcal{M}$, or a good approximation to it, without knowing x^*.

HOW? Optimality conditions for the problem $\min_{x\in\Omega} f(x)$ are

$$x^* - \nabla f(x^*) \in x^* + N_\Omega(x^*),$$

where $N_\Omega(x)$ is the normal cone to Ω at x:

$$N_\Omega(x) = \{s \,|\, s^T(z-x) \leq 0 \text{ for all } z \in \Omega\}.$$

When $\mathcal{M}$ is defined appropriately, we find that

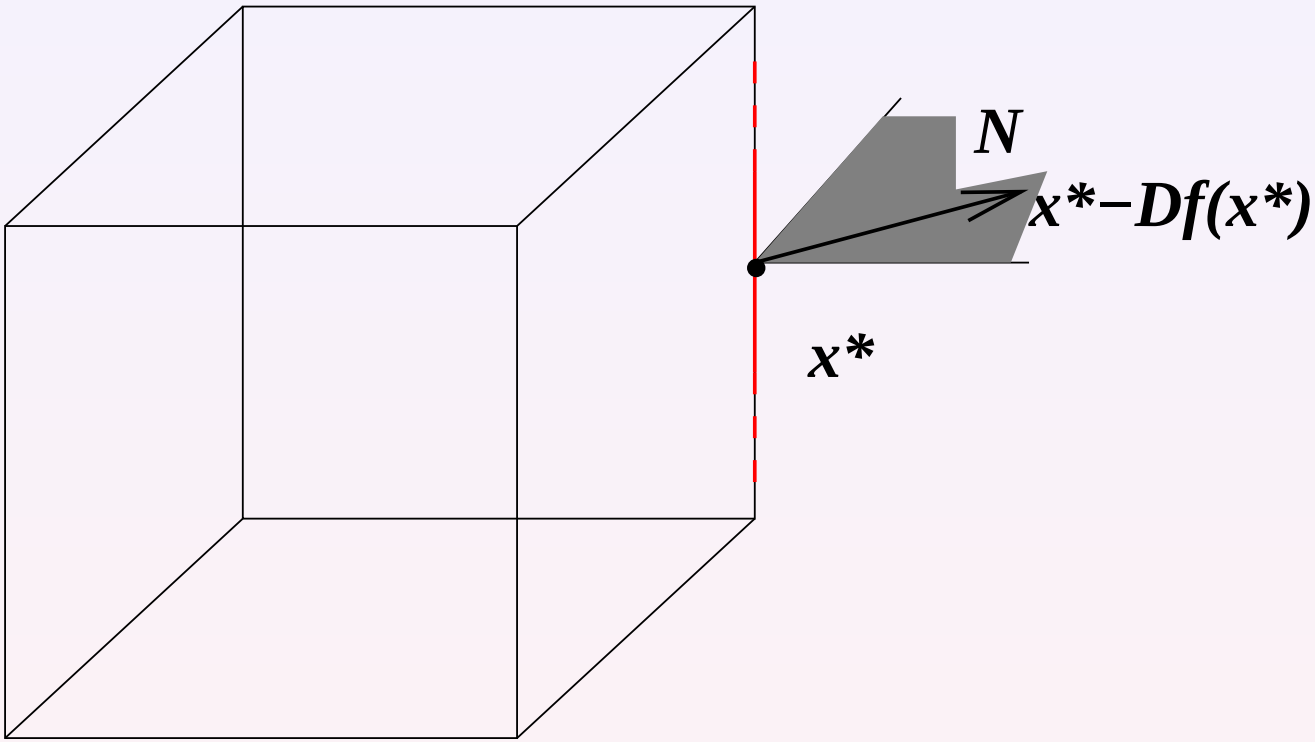
$$R(\Omega, \mathcal{M}) := \{x + s \,|\, x \in \mathcal{M} \text{ and } s \in N_\Omega(x)\}$$

has an interior in $\mathbb{R}^n$. If x^* satisfies a nondegeneracy condition — that is, $-\nabla f(x^*) \in \text{ri}\, N_\Omega(x^*)$ (relative interior), we have

$$x - \nabla f(x) \in R(\Omega, \mathcal{M})$$

for all x sufficiently close to x^*. From such points, projection recovers $\mathcal{M}$:

$$P_\Omega(x - \nabla f(x)) \in \mathcal{M}.$$


N
x*−Df(x*)
x*

The non-averaged iterates from gradient projection methods eventually lie on the correct manifold $\mathcal{M}$. The same is true for dual-averaging methods (where the gradient term is averaged over all steps).

When we have a nonsmooth regularizer ψ, instead of Ω, the analogous property is that the solution of the shrink subproblem, for some fixed positive α, lies on the optimal manifold $\mathcal{M}$.

Under reasonable conditions on α_k, the "basic" shrink method eventually has all its iterates on $\mathcal{M}$. Also true for dual-averaged methods.

In practice, often use heuristics for deciding when $\mathcal{M}$ (or a small superset) has been reached. If a distance-to-solution bound is available, and if Lipschitz constant for ∇f is known, can make this decision more rigorous.

How Might This Help?

Consider again logistic regression with regularizer $\psi(w) = \|w\|_1$.

$$\mathcal{L}(w) = -\frac{1}{m}\left[\sum_{y_i=-1} w^T x_i - \sum_{i=1}^{m} \log(1 + e^{w^T x_i})\right].$$

Requires calculation of Xw where $X = [x_i^T]_{i=1}^m$. (Can be cheap if w has few nonzeros.) For gradient, have

$$\nabla \mathcal{L}(w) = \frac{1}{m} X^T u, \quad \text{where} \quad u_i = \begin{cases} -(1 + e^{w^T x_i})^{-1}, & y_i = -1, \\ (1 + e^{-w^T x_i})^{-1}, & y_i = +1. \end{cases}$$

requires m exponentials, and a matrix-vector multiply by X (with a *full* vector u).

If just a subset $\mathcal{G}$ of components needed, multiply by a column submatrix $X_{\cdot\mathcal{G}}^T$ — much cheaper than full gradient if $|\mathcal{G}| \ll n$.

Reduced Hessian

$$\nabla^2 \mathcal{L}(w) = \frac{1}{n} X^T \text{diag}(v) X, \quad \text{where} \quad v_i = \frac{e^{w^T x_i}}{(1 + e^{w^T x_i})^2}.$$

Often much cheaper to calculate $|\mathcal{G}| \times |\mathcal{G}|$ reduced Hessian than the full Hessian.

Can use *sampling* (Byrd et al., 2010) to approximate the projected Hessian: take a subset $\mathcal{S} \subset \{1, 2, \ldots, m\}$ and use $X_{\mathcal{S}\mathcal{G}}$ in place of $X_{\cdot\mathcal{G}}$. Reduces evaluation cost by a factor $|\mathcal{S}|/m$.

Higher-Order Shrinking Method

for problem $\min_x f(x) + \tau\|x\|_1$. Step k:

- Choose a subset $\mathcal{G}_k \supset \{i \,|\, x_k(i) \neq 0\}$
- Evaluate $\nabla_{\mathcal{G}_k} f(x_k)$ and solve (in closed form):

$$\min_d \nabla f(x_k)^T d + \frac{1}{2\alpha_k} d^T d + \tau\|x_k + d\|_1, \;\; \text{s.t.} \;\; d(i) = 0 \;\; \text{for } i \notin \mathcal{G}_k.$$

- Repeat with a decreasing sequence of α_k, until sufficient decrease; Set $x_k^+ = x_k + d$;
- Define $\mathcal{C}_k \subset \mathcal{G}_k$ by $\mathcal{C}_k := \{i \,|\, x_k^+(i) \neq 0\}$.
- Calculate reduced Newton step in $\mathcal{C}_k$ components with (approximate) reduced Hessian matrix $H_{\mathcal{C}_k\mathcal{C}_k}$ and reduced gradient $\nabla_{\mathcal{C}_k} f$, evaluated at x_k^+. Take this reduced step if is gives an improvement in objective, giving x_{k+1}. Otherwise, settle for $x_{k+1} \leftarrow x_k^+$.

Can be generalized to other separable regularizers $\psi(x)$. Choice of subset must conform to separation in regularizer, and all components must be checked periodically.

Newton-Like Methods

Newton-like methods are ubiquitous in smooth optimization — motivated by second-order Taylor series.

(Basic) Newton's Method steps obtained from

$$x_{k+1} = \arg\min_z \; f(x_k) + \nabla f(x_k)^T (z - x_k) + \frac{1}{2}(z - x_k)^T H_k (z - x_k),$$

where $H_k = \nabla^2 f(x_k)$. Near a local minimizer with second-order sufficient conditions, converges superlinearly: $\|x_{k+1} - x^*\| = o(\|x_k - x^*\|)$.

Can modify by

- adding a prox-term e.g. multiple of $\|z - x_k\|^2$;
- Adding a trust-region constraint $\|z - x_k\| \leq \Delta_k$ (equivalent);
- doing a line search along d.

Choices of H_k

Hessian $\nabla^2 f$ often expensive to evaluate, so can use *approximations*, e.g.

- Re-use $\nabla^2 f$ from a previous iterate.
- Use a sampled approximation to $\nabla^2 f(x_k)$ (see above).
- Use a diagonal approximation — at least gets the scaling right. See e.g. Barzilai-Borwein above.
- quasi-Newton methods (BFGS, L-BFGS), which define H_k to be a matrix that mimics the bevavior of the true Hessian over previous steps. Requires only gradients ∇f.
- Other approximations that exploit the structure of the problem. e.g. for nonlinear least squares $f(x) = (1/2)\|r(x)\|_2^2$ for $r : \mathbb{R}^n \to \mathbb{R}^m$, Hessian is

$$\nabla^2 f(x) = J(x)^T J(x) + (1/2) \sum_{i=1}^{m} r_i(x) \nabla^2 r_i(x),$$

where J is the $m \times n$ Jacobian of r. In Gauss-Newton method, use $H(x) = J(x)^T J(x)$.

Higher-Order Information and Constraints

Higher-order methods can be extended to presence of constraints $x \in \Omega$ or regularizers $\psi(x)$ provided these elements can be incorporated explicitly into the subproblems. e.g. for constraints

$$x_{k+1} = \arg\min_{z \in \Omega} f(x_k) + \nabla f(x_k)^T (z - x_k) + \frac{1}{2}(z - x_k)^T H_k (z - x_k),$$

and for regularizers

$$x_{k+1} = \arg\min_{z} f(x_k) + \nabla f(x_k)^T (z - x_k) + \frac{1}{2}(z - x_k)^T H_k (z - x_k) + \tau\psi(z).$$

These subproblems are typically harder to solve than the "shrink" subproblem, unless H_k is simple (e.g. diagonal).

In practice, can do manifold identification and reduction (see above), or form simpler approximations to Ω (but then may need to incorporate curvature information about Ω into H_k to ensure fast convergence).

Solving for x_{k+1}

When H_k positive definite, can solve Newton equations explicitly by solving

$$H_k(z - x_k) = -\nabla f(x_k).$$

Alternatively, apply conjugate gradient to this system to get an inexact solution. Each iterate requires a multiplication by H_k.

Can precondition CG, e.g. by using sample approximations, structured approximations, or Krylov subspace information gathered at previous evaluations of $\nabla^2 f$.

L-BFGS stores H_k in implicit form, by means of $2m$ vectors in $\mathbb{R}^n$, for a small parameter m (e.g. 5). Recovers solution of the equation above via $2m$ inner products.

"Higher-Order" Methods in ML

Several approaches tried (in addition to sampling and reduced-space techniques discussed above).

- Bordes et al. (2009, corrected 2010) for SVM

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

  scales the stochastic gradient step with a diagonal H_k, obtained from finite differences of the last estimated gradient over the last step.
- Schraudolph et al. (AISTATS 2007) "online BFGS" uses conventional quasi-Newton update formulae (e.g. L-BFGS) based on estimated gradient differences over previous steps.

Since the gradients are so inexact (based on just one data point), both in update and right-hand side of the step equations, these methods are really stochastic gradient with interesting scaling, rather than quasi-Newton in the conventional sense.

Further Reading

1. L. Bottou and A. Moore, "Learning with large datasets," Tutorial at NIPS 2007 (slides on web).
2. J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd edition, Springer, 2006.
3. R. H. Byrd, G. M. Chin, W. Neveitt, and J. Nocedal. "On the use of stochastic hessian information in unconstrained optimization." Technical Report, Northwestern University, June 2010.
4. M. Fisher, J. Nocedal, Y. Tremolet, and S. J. Wright. "Data assimilation in weather forecasting: A case study in PDE-constrained optimization." Optimization and Engineering, 10, pp. 409–426, 2009.
5. A. S. Lewis and S. J. Wright. "Identifying activity." Technical report, ORIE, Cornell University. Revised April 2010.
6. S. J. Wright, "Accelerated block-coordinate relaxation for regularized optimization," Technical Report, August 2010.
7. W. Hare and A. Lewis. "Identifying active constraints via partial smoothness and prox-regularity." Journal of Convex Analysis, 11, pp. 251–266, 2004.

V. Decomposition / Coordinate Relaxation

For min $f(x)$, at iteration k, choose a subset $\mathcal{G}_k \subset \{1, 2, \ldots, n\}$ and take a step d_k only in these components. i.e. fix $d_k(i) = 0$ for $i \notin \mathcal{G}_k$.

Gives more manageable subproblem sizes, in practice.

Can

- take a reduced gradient step in the $\mathcal{G}_k$ components;
- take multiple "inner iterations"
- actually solve the reduced subproblem in the space defined by $\mathcal{G}_k$.

Constraints and Regularizers Complicate Things

For $\min_{x \in \Omega} f(x)$, need to put enough components into $\mathcal{G}_k$ to stay feasible, as well as make progress.

Example: $\min f(x_1, x_2)$ with $x_1 + x_2 = 1$. Relaxation with $\mathcal{G}_k = \{1\}$ or $\mathcal{G}_k = \{2\}$ won't work.

For separable regularizer (e.g. Group LASSO) with

$$\psi(x) = \sum_{g \in G} \psi_g(x_{[g]}),$$

need to ensure that $\mathcal{G}_k$ is a union of the some index subsets $[g]$. i.e. the relaxation components must be consonant with the partitioning.

Decomposition and Dual SVM

Decomposition has long been popular for solving the dual (QP) formulation of SVM, since the number of variables (= number of training examples) is sometimes very large.

SMO: Each G_k has two components.

LIBSVM: SMO approach (still $|\mathcal{G}_k| = 2$), with different heuristic for choosing $\mathcal{G}_k$.

LASVM: Again $|\mathcal{G}_k| = 2$, with focus on online setting.

SVM-light: Small $|\mathcal{G}_k|$ (default 10).

GPDT: Larger $|\mathcal{G}_k|$ (default 400) with gradient projection solver as inner loop.

Choice of $\mathcal{G}_k$ and Convergence Results

Some methods (e.g. Tseng and Yun, 2010) require $\mathcal{G}_k$ to be chosen so that *the improvement in subproblem objective obtained over the subset $\mathcal{G}_k$ is at least a fixed fraction of the improvement available over the whole space*. Undesirable, since to check it, usually need to evaluate the **full gradient** $\nabla f(x_k)$.

Alternative is a *generalized Gauss-Seidel* requirement, where each coordinate is "touched" at least once every T iterations:

$$\mathcal{G}_k \cup \mathcal{G}_{k+1} \cup \ldots \cup \mathcal{G}_{k+T-1} = \{1, 2, \ldots, n\}.$$

Can show global convergence (e.g. Tseng and Yun, 2009; Wright, 2010).

There are also results on

- global linear convergence rates
- optimal manifold identification
- fast local convergence for an algorithm that takes reduced steps on the estimated optimal manifold.

All *deterministic* analyses.

Stochastic Coordinate Descent

Analysis tools of stochastic gradient may be useful. If steps have the form $x_{k+1} = x_k - \alpha_k g_k$, where

$$g_k(i) = \begin{cases} [\nabla f(x_k)]_i & \text{if } i \in \mathcal{G}_k \\ 0 & \text{otherwise,} \end{cases}$$

With suitable random selection of $\mathcal{G}_k$ can ensure that g_k (appropriately scaled) is an unbiased estimate of $\nabla f(x_k)$. Hence can apply SGD techniqes discussed earlier, to choose α_k and obtain convergence.

Nesterov (2010) proposes another randomized approach for the unconstrained problem with known separate Lipschitz constants L_i:

$$\left\| \frac{\partial f_i}{\partial x_i}(x + he_i) - \frac{\partial f_i}{\partial x_i}(x) \right\| \leq L_i |h|, \quad i = 1, 2, \ldots, n.$$

(Works with *blocks* too, instead of individual components.)

At step k:

- Choose index $i_k \in \{1, 2, \ldots, n\}$ with probability $p_i := L_i / (\sum_{j=1}^n L_j)$;
- Take gradient step in i_k component:

$$x_{k+1} = x_k - \frac{1}{L_{i_k}} \frac{\partial f}{\partial x_{i_k}} e_{i_k}.$$

Basic convergence result:

$$E[f(x_k)] - f^* \leq \frac{C}{k}.$$

As for SA (earlier) but without any strong convexity assumption.

Can also get **linear** convergence results (in expectation) by assuming strong convexity in f, according to different norms.

Can also **accelerate** in the usual fashion (see above), to improve expected convergence rate to $O(1/k^2)$.

Further Reading

1. P. Tseng and S. Yun, "A coordinate gradient descent method for linearly constrained smooth optimization and support vector machines training." Computational Optimization and Applications, 47, pp. 179–206, 2010.
2. P. Tseng and S. Yun, "A coordinate gradient descent method for nonsmooth separable minimization." Mathematical Programming, Series B, 117. pp. 387–423, 2009.
3. S. J. Wright, "Accelerated block-coordinate relaxation for regularized optimization." Technical Report, UW-Madison, August 2010.
4. Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems." CORE Discussion Paper 2010/2, CORE, UCL, January 2010.

Conclusions

We've surveyed a number of topics in algorithmic fundamentals, with an eye on recent developments, and on topics of relevance (current or future) to machine learning.

The talk was far from exhaustive. Literature on Optimization in ML is huge and growing.

There is much more to be gained from the interaction between the two areas.

FIN