Network Optimization
Chapter 2: Mathematical Tools

Yiyin Wang
yiyinwang@sjtu.edu.cn
Department of Automation
Shanghai Jiao Tong University

Nov. 4, 2014

Acknowledgements

Slides borrow heavily from lectures by Stephen Boyd (Stanford), Lieven Vandenberghe (UCLA), and Mung Chiang (Princeton)

Part 3: Algorithms

- 2.4 Descent algorithm
- 2.5 Interior-point algorithm

Unconstrained minimization

- Given $f : \mathbf{R}^n \to \mathbf{R}$ convex and twice differentiable:

$$\text{minimize} \qquad f(x)$$

  Optimizer x^*. Optimized value $p^* = f(x^*)$
- Necessary and sufficient condition of optimality:

$$\bigtriangledown f(x^*) = 0$$

Unconstrained minimization

- Iterative algorithm: computes a sequence of points $x^{(k)} \in \mathrm{dom}\ f,\ \ k = 0, 1, \dots,$ such that

$$\lim_{k \to \infty} f(x^{(k)}) = p^*$$

- Terminate algorithm when $f(x^{(k)}) - p^* \leq \epsilon$ for a specified $\epsilon > 0$
- Can be interpreted as iterative methods for solving optimality condition

$$\bigtriangledown f(x^*) = 0$$

- The iterative algorithm require a starting point $x^{(0)}$ such that
  - $x^{(0)} \in \mathrm{dom}\ f$
  - Sublevel set $S = \{x \,|\, f(x) \leq f(x^{(0)})\}$ is closed

Examples

- Least-squares: minimize

$$\|Ax - b\|_2^2 = x^T(A^TA)x - 2(A^Tb)^Tx + b^Tb$$

Optimality condition ($\bigtriangledown f(x^*) = 0$) is system of linear equations:

$$A^TAx^* = A^Tb$$

called normal equations for least squares

- Unconstrained geometric programming: minimize

$$f(x) = \log\left(\sum_{i=1}^{m} \exp(a_i^Tx + b_i)\right)$$

Optimality condition ($\bigtriangledown f(x^*) = 0$) has no analytic solution:

$$\bigtriangledown f(x^*) = \frac{1}{\sum_{j=1}^{m} \exp(a_j^Tx^* + b_j)} \sum_{i=1}^{m} \exp(a_i^Tx^* + b_i)a_i = 0$$

Strong convexity

- f assumed to be strongly convex: there exits $m > 0$ such that

$$\bigtriangledown^2 f(x) \succeq mI$$

which also implies that there exists $M \geq m$ such that

$$\bigtriangledown^2 f(x) \preceq MI$$

- Bound optimal value:

$$f(x) - \frac{1}{2m}\| \bigtriangledown f(x)\|_2^2 \leq p^* \leq f(x) - \frac{1}{2M}\| \bigtriangledown f(x)\|_2^2$$

- Suboptimality condition:

$$\| \bigtriangledown f(x)\|_2^2 \leq (2m\epsilon)^{1/2} \to f(x) - p^* \leq \epsilon$$

- Distance between x and optimal x^*

$$\|x - x^*\|_2 \leq \frac{2}{m}\| \bigtriangledown f(x)\|_2$$

Descent Methods

- Minimizing sequence $x^{(k)}$, $k = 1, \ldots,$ (where $t^{(k)} > 0$)

$$x^{(k+1)} = x^{(k)} + t^{(k)} \triangle x^{(k)}$$

  - $\triangle x^{(k)}$: the search direction or step
  - $t^{(k)}$: the step length or step size

- Descent methods:

$$f(x^{(k+1)}) < f(x^{(k)})$$

  By convexity of f, search direction must make an acute angle with negative gradient:

$$\triangledown f(x^{(k)})^T \triangle x^{(k)} < 0$$

  Because otherwise, $f(x^{(k+1)}) \geq f(x^{(k)})$ since
  $f(x^{(k+1)}) \geq f(x^{(k)}) + \triangledown f(x^{(k)})^T (x^{(k+1)} - x^{(k)})$

General descent method

Given a starting point $x^{(0)} \in \mathrm{dom}\ f$

Repeat

1. Determine a descent direction $\triangle x^{(k)}$
2. Line search: choose a step size $t > 0$
3. Update: $x^{(k+1)} = x^{(k)} + t\triangle x^{(k)}$

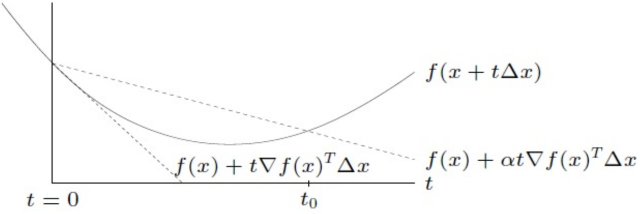
Until stopping criterion satisfied

Line search types

- Exact line search: $t = \text{argmin}_{s \geq 0} f(x + s\triangle x)$
- Backtracking line search (with parameters $\alpha \in (0, 1/2), \beta \in (0, 1))$
  - Starting at $t = 1$, repeat $t := \beta t$ until

$$f(x + t\triangle x) < f(x) + \alpha t \bigtriangledown f(x)^T \triangle x$$

  - graphical interpretation: backtrack until $t \leq t_0$



Gradient descent method

Given a starting point $x \in \mathrm{dom}\ f$
Repeat

1. $\triangle x := -\bigtriangledown f(x)$
2. Line search: choose a step size $t > 0$
3. Update: $x := x + t\triangle x$

Until stopping criterion satisfied

- Stopping criterion usually of the form $\| \bigtriangledown f(x)\|_2 \leq \epsilon$
- Convergence result: for strong convex f,

$$f(x^{(k)}) - p^* \leq c^k(f(x^{(0)}) - p^*)$$

  $c \in (0, 1)$ depends on $m, x^{(0)}$, line search type
- Very simple, but often very slow; rarely used in practice

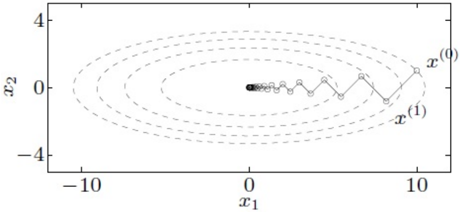
Examples: quadratic problem in $\mathbf{R}^2$

$$\text{minimize } f(x) = \frac{1}{2}(x_1^2 + \gamma x_2^2) \quad (\gamma > 0)$$

with exact line search, starting at $x^{(0)} = (\gamma, 1)$:

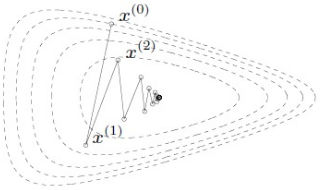
$$x_1^{(k)} = \gamma \left(\frac{\gamma - 1}{\gamma + 1}\right)^k, x_2^{(k)} = \left(-\frac{\gamma - 1}{\gamma + 1}\right)^k$$

- Very slow if $\gamma \gg 1$ or $\gamma \ll 1$
- Example for $\gamma = 10$

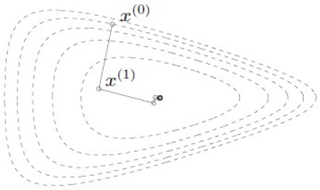


Examples: nonquadratic example

$$f(x_1, x_2) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1}$$



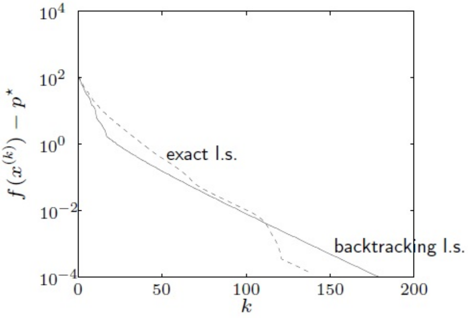
Backtracking line search



Exact line search

Examples: a problem in $\mathbf{R}^{100}$

$$f(x) = c^T x - \sum_{i=1}^{500} \log(b_i - a_i^T x)$$



'Linear' convergence, i.e. a straight line on a semilog plot

Steepest descent method

- Normalized steepest descent direction (at x, for norm $\|\cdot\|$):

$$\triangle x_{\mathrm{nsd}} = \mathrm{argmin}\{\triangledown f(x)^T \nu \mid \|\nu\| = 1\}$$

  Interpretation: for small ν, $f(x+\nu) \approx f(x) + \triangledown f(x)^T \nu$; direction $\triangle x_{\mathrm{nsd}}$ is unit-norm step with most negative directional derivative

- (unnormalized) steepest descent direction

$$\triangle x_{\mathrm{sd}} = \| \triangledown f(x)\|_* \triangle x_{\mathrm{nsd}}$$

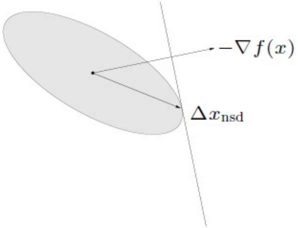
  satisfies $\triangledown f(x)^T \triangle_{\mathrm{sd}} = -\| \triangledown f(x)\|_*^2$

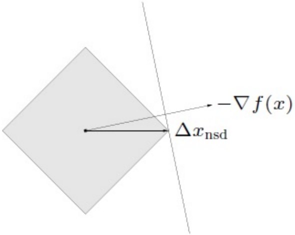
- Steepest decent method
  - General descent method with $\triangle x = \triangle x_{\mathrm{sd}}$
  - Convergence properties similar to gradient descent

Examples

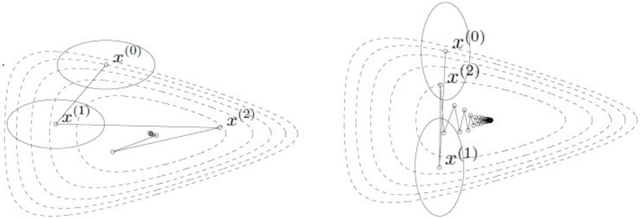
- Euclidean norm: $\triangle x_{\mathrm{sd}} = -\bigtriangledown f(x)$
- Quadratic norm
  $\|x\|_P = (x^T P x)^{1/2} (P \in \mathbf{S}^n_{++}) : \triangle x_{\mathrm{sd}} = -P^{-1} \bigtriangledown f(x)$
- ℓ_1-norm: $\triangle x_{\mathrm{sd}} = -(\partial f(x)/\partial x_i) e_i$, where
  $|\partial f(x)/\partial x_i| = \| \bigtriangledown f(x)\|_\infty$

Unit balls and normalized steepest descent directions for a quadratic norm and the ℓ_1-norm





Choice of norm for steepest descent



- Steepest descent with backtracking line search for two quadratic norms
- Ellipses show $\{x \mid \|x - x^{(k)}\|_P = 1\}$
- Equivalent interpretation of steepest descent with quadratic norm $\|\cdot\|_P$: gradient descent after change of variables $\bar{x} = P^{1/2}x$
  Show choice of P has strong effect on speed of convergence

Newton step

$$\triangle x_{\mathrm{nt}} = -\bigtriangledown^2 f(x)^{-1} \bigtriangledown f(x)$$

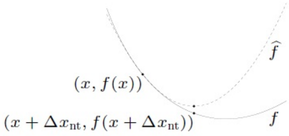
Interpretations

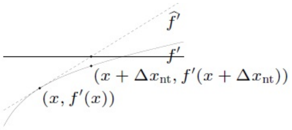
- $x + \triangle x_{\mathrm{nt}}$ minimizes second order approximation

$$\hat{f}(x+\nu) = f(x) + \bigtriangledown f(x)^T \nu + \frac{1}{2}\nu^T \bigtriangledown^2 f(x)\nu$$

- $x + \triangle x_{\mathrm{nt}}$ solves linearized optimality condition

$$\bigtriangledown f(x+\nu) \approx \bigtriangledown \hat{f}(x+\nu) = \bigtriangledown f(x) + \bigtriangledown^2 f(x)\nu = 0$$





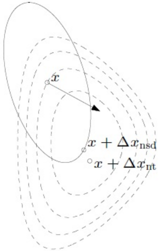
Newton step

- $\triangle x_{\mathrm{nt}}$ is independent of linear (or affine) changes of coordinates. It is affine invariant

$$x = Ty, \quad x + \triangle x_{\mathrm{nt}} = T(y + \triangle y_{\mathrm{nt}})$$

- $\triangle x_{\mathrm{nt}}$ is steepest descent direction at x in local Hessian norm

$$\|u\|_{\nabla^2 f(x)} = (u^T \nabla^2 f(x) u)^{1/2}$$



Dash lines are contour lines of f; ellipse is $\{x + \nu \,|\, \nu^T \nabla^2 f(x)\nu = 1\}$ arrow shows $-\nabla f(x)$

Newton decrement

$$\lambda(x) = (\nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x))^{1/2}$$

a measure of the proximity of x to x^*

Properties

- Gives an estimate of $f(x) - p^*$, using quadratic approximation $\hat{f}$:

$$f(x) - \inf_y \hat{f}(y) = f(x) - \hat{f}(x + \triangle x_{\mathrm{nt}}) = \frac{1}{2}\lambda(x)^2$$

- Equal to the norm of the Newton step in the quadratic Hessian norm

$$\lambda(x) = (\triangle x_{\mathrm{nt}}^T \nabla^2 f(x) \triangle x_{\mathrm{nt}})^{1/2}$$

- Directional derivative in the Newton direction, $\nabla f(x)^T \triangle x_{\mathrm{nt}} = -\lambda(x)^2$, can be used in the backtracking line search
- Affine invariant (unlike $\| \nabla f(x) \|_2$)

Newton's method

- Given a starting point $x \in \mathrm{dom}\ f$, tolerance $\epsilon > 0$
- Repeat
  1. Compute the Newton step and decrement

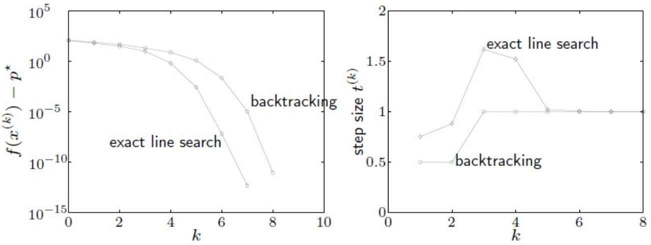
     $$\triangle x_{\mathrm{nt}} := -\bigtriangledown^2 f(x)^{-1} \bigtriangledown f(x); \quad \lambda^2 := \bigtriangledown f(x)^T \bigtriangledown^2 f(x)^{-1} \bigtriangledown f(x)$$

  2. Stopping criterion. Quit if $\lambda^2/2 \le \epsilon$
  3. Line search. Choose step size t by backtracking line search.
  4. Update. $x := x + t \triangle x_{\mathrm{nt}}$

Advantages of Newton method: fast, robust, scalable

Example in $\mathbf{R}^{100}$

$$f(x) = c^T x - \sum_{i=1}^{500} \log(b_i - a_i^T x)$$



- Backtracking parameters $\alpha = 0.01, \beta = 0.5$
- Backtracking line search almost as fast as exact l.s. (and much simpler)
- Clearly shows two phases in algorithm

Summary about Newton's method

- Convergence of Newton's method is rapid in general, and quadratic near x^*
- Newton's method is affine invariant
- Newton's method scales well with problem size
- The good performance of Newton's method is not dependent on the choice of algorithm parameters
- The main disadvantage of Newton's method is the cost of forming and storing the Hessian ($\bigtriangledown^2 f(x)$), and the cost of computing the Newton step