COMP 9602: Convex Optimization

Algorithms for Unconstrained Optimization

Dr. C Wu

Department of Computer Science
The University of Hong Kong

Roadmap

Theory	convex set convex function standard forms of optimization problems, quasi-convex optimization linear program, integer linear program quadratic program geometric program semidefinite program vector optimization duality
Algorithm	unconstrained optimization equality constrained optimization interior-point method localization methods subgradient method decomposition methods

Unconstrained optimization

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

- f convex, twice continuously differentiable (hence $\mathbf{dom}\, f$ open)
- we assume optimal value $p^\star = \inf_x f(x)$ is attained (and finite)

solving the unconstrained optimization <=> find the solution to:

$$\nabla f(x^*) = 0 \qquad \text{(n equations)}$$

- usually need iterative algorithms:
  - produce a sequence of $x^{(k)} \in dom(f), k = 1, 2, \ldots$ with $f(x^{(k)}) \to \min f(x)$ (i.e., $\nabla f(x^{(k)}) \to 0$) when $k \to \infty$
  - the algorithm terminates when $f(x^{(k)}) - \min f(x) \leq \epsilon$ where $\epsilon > 0$ or $\| \nabla f(x^{(k)}) \|_2 \leq \epsilon$

Descent methods

- start with $x^{(0)}$
- update $x^{(k+1)} = x^{(k)} + t^{(k)} \triangle x^{(k)}$ such that $f(x^{(k+1)}) < f(x^{(k)})$
  - other notations: $x^+ = x + t\Delta x$, $x := x + t\Delta x$
  - Δx is the *step*, or *search direction*; t is the *step size*, or *step length*
  - from convexity, $f(x^+) < f(x)$ implies $\nabla f(x)^T \Delta x < 0$ ($i.e.$, Δx is a *descent direction*)

General descent method.

given a starting point $x \in \mathbf{dom}\, f$.
repeat
1. Determine a descent direction Δx.
2. *Line search.* Choose a step size $t > 0$.
3. *Update.* $x := x + t\Delta x$.

until stopping criterion is satisfied.

Descent methods

- Determine a descent direction
  - one approach is gradient descent

    $$\triangle x = - \bigtriangledown f(x)$$

- Choose a step size
  - line search methods

    exact line search: $t = \mathrm{argmin}_{t>0} f(x + t\Delta x)$

    backtracking line search (with parameters $\alpha \in (0, 1/2)$, $\beta \in (0, 1)$)

    - starting at $t = 1$, repeat $t := \beta t$ until

    $$f(x + t\Delta x) \leq f(x) + \alpha t \nabla f(x)^T \Delta x$$

Gradient descent method

- Gradient descent method:

  general descent method with $\Delta x = -\nabla f(x)$

given a starting point $x \in \mathbf{dom}\, f$.
repeat
1. $\Delta x := -\nabla f(x)$.
2. *Line search.* Choose step size t via exact or backtracking line search.
3. *Update.* $x := x + t\Delta x$.

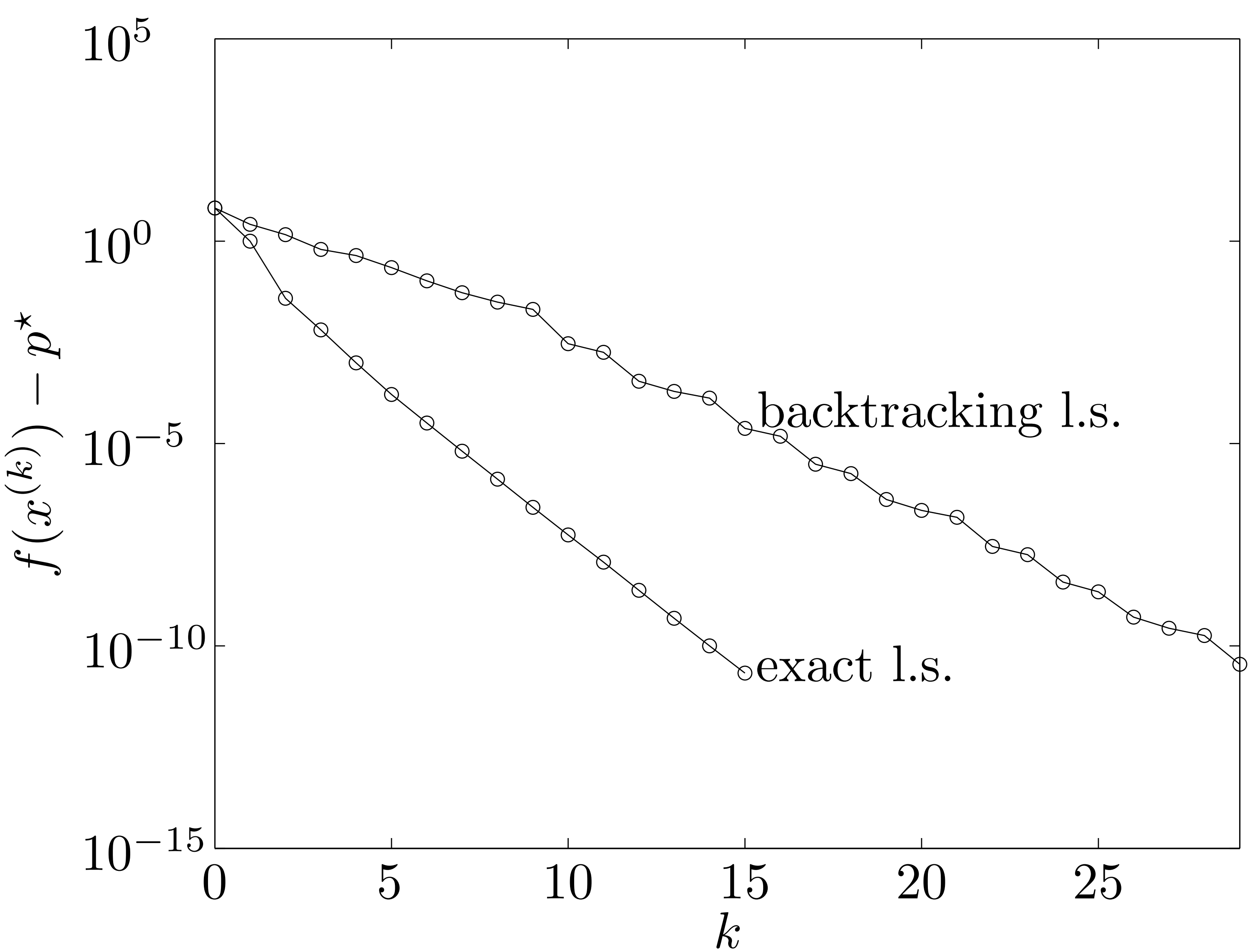
until stopping criterion is satisfied.

Gradient descent method

- Remarks
  - can be very slow (linear convergence); rarely used in practice

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

linear convergence: the error converges to zero at least as fast as a geometric series



Gradient descent method

- find the steepest descent direction — steepest descent method

  normalized steepest descent direction

  $$\Delta x_{\text{nsd}} = \text{argmin}\{\nabla f(x)^T v \mid \|v\| = 1\}$$

  $\triangle x = -\bigtriangledown f(x)$ is the steepest descent direction with respect to Euclidean norm

  so performance of steepest descent similar to gradient descent

Newton's method

Newton's direction $\Delta x_{\mathrm{nt}} = -\nabla^2 f(x)^{-1} \nabla f(x)$

- One interpretation
  - $x + \Delta x_{\mathrm{nt}}$ minimizes second order approximation

$$\widehat{f}(x+v) = f(x) + \nabla f(x)^T v + \frac{1}{2} v^T \nabla^2 f(x) v$$

intuition: f twice differentiable, so this quadratic model is very accurate when x is near x*.

Newton's method

Newton's decrement (Stopping criteria):

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

a measure of the proximity of x to $x^\star$

- gives an estimate of $f(x) - p^\star$, using quadratic approximation $\widehat{f}$:

$$f(x) - \inf_y \widehat{f}(y) = \frac{1}{2}\lambda(x)^2$$

- directional derivative in the Newton direction: $\nabla f(x)^T \Delta x_{\mathrm{nt}} = -\lambda(x)^2$

Newton's method

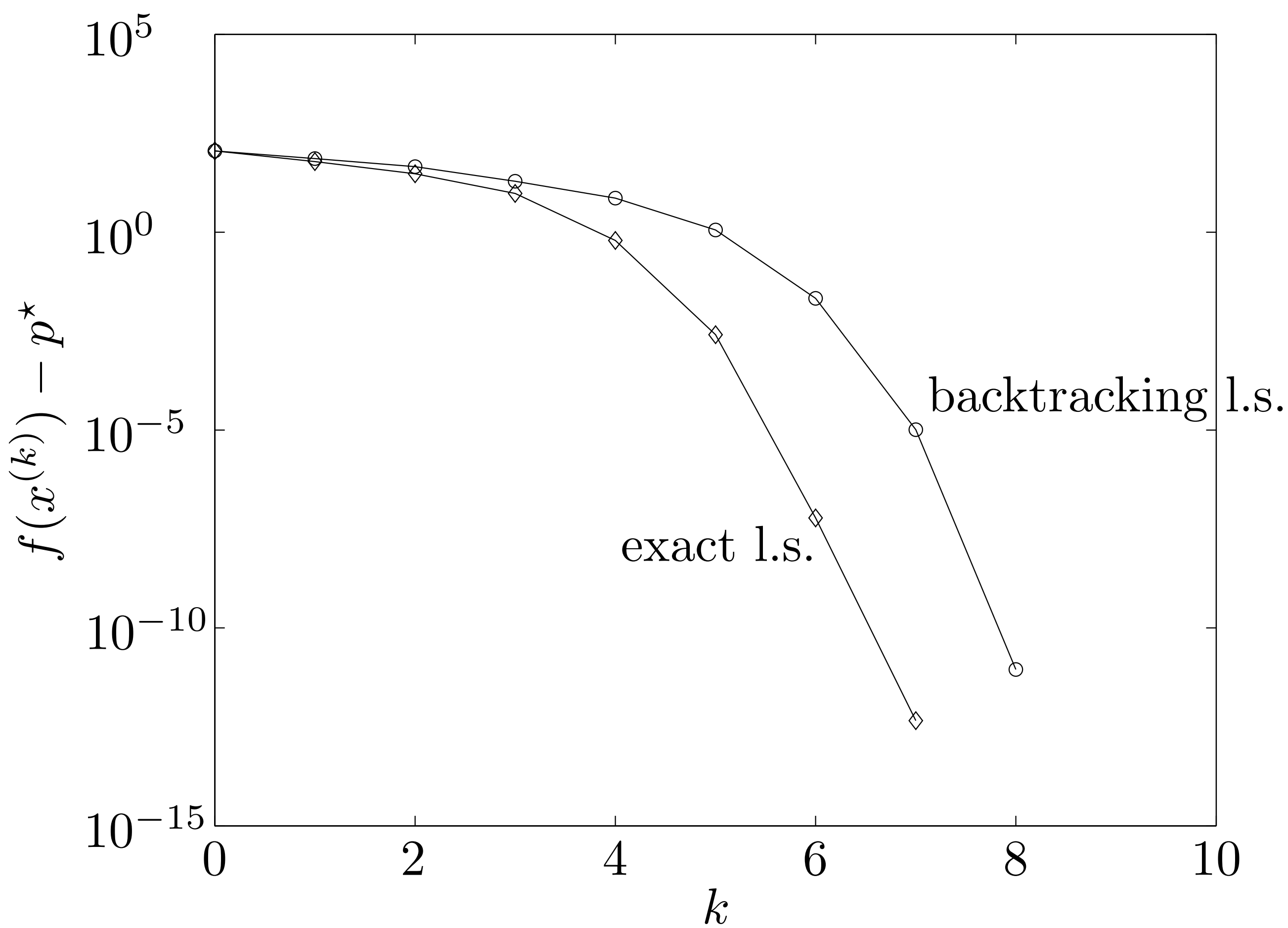
- (damped or guarded) Newton's method:

given a starting point $x \in \mathbf{dom}\, f$, tolerance $\epsilon > 0$.
repeat
1. *Compute the Newton step and decrement.*
   $$\Delta x_{\mathrm{nt}} := -\nabla^2 f(x)^{-1} \nabla f(x); \quad \lambda^2 := \nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x).$$
2. *Stopping criterion.* **quit** if $\lambda^2/2 \leq \epsilon$.
3. *Line search.* Choose step size t by backtracking line search.
4. *Update.* $x := x + t\Delta x_{\mathrm{nt}}$.

pure Newton's method: t=1

Newton's method

- Converge in 2 phases
  - damped Newton phase: t<1 (most iterations require backtracking steps)
  - quadratically convergent phase: t=1 ($\| \nabla f(x) \|_2$ converges to 0 quadratically, i.e., the number of correct digits doubles at each iteration)



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

with $m = 500$ terms and $n = 100$ variables.

- Convergence is rapid in general and quadratic near x*
- Cost: compute and store $\nabla^2 f(x)$, and $\nabla^2 f(x)^{-1}$

- Reference
  - Chapter 9.1-9.5, Convex Optimization.
- Acknowledgement
  - Some materials are extracted from the slides created by Prof. Stephen Boyd for the textbook
  - Some materials are extracted from the lecture notes of Convex Optimization by Prof. Wei Yu at the University of Toronto