Unconstrained Minimization

Lecture 11, Nonlinear Programming

National Taiwan University

December 20, 2016

Table of contents

Unconstrained minimization (1/2)

- In this lecture, we discuss methods for solving the unconstrained optimization problem

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

  where $f : \mathbf{R}^n \to \mathbf{R}$ is convex and twice continuously differentiable (which implies that **dom** f is open).
- We will assume that the problem is solvable, i.e., there exists an optimal point x^*.
- We denote the optimal value, $\inf_x f(x) = f(x^*)$, as p^*.
- Since f is differentiable and convex, a necessary and sufficient condition for a point x^* to be optimal is $\nabla f(x^*) = 0$.

Unconstrained minimization (2/2)

- Thus, solving the unconstrained minimization problem minimize $f(x)$ is the same as finding a solution of

$$\nabla f(x^*) = 0,$$

which is a set of n equations in the n variables $x_1, \ldots, x_n$.

- We sometimes can find an analytical solution for $\nabla f(x^*) = 0$, but in general it must be solved by an iterative algorithm that computes a sequence of points $x^{(0)}, x^{(1)}, \ldots \in \textbf{dom}\ f$ with $f(x^{(k)}) \to p^*$ as $k \to \infty$.
- Such a sequence of points is called a **minimizing sequence** for the problem minimize $f(x)$.
- The algorithm is terminated when $f(x^{(k)}) - p^* \leq \epsilon$, where $\epsilon > 0$ is some specified tolerance.

Initial point and sublevel set

- The iterative methods generally require a suitable starting point $x^{(0)} \in \mathbf{dom}\ f$.
- In addition, the sublevel set

$$S = \left\{x \in \mathbf{dom}\ f \mid f(x) \leq f(x^{(0)})\right\}$$

  must be closed.
- A function f is said to be **closed** if all its sublevel sets are closed.
  - Continuous functions with $\mathbf{dom}\ f = \mathbf{R}^n$ are closed, so if $\mathbf{dom}\ f = \mathbf{R}^n$, the initial sublevel set condition is satisfied by any $x^{(0)}$.
  - Another important class of closed functions are continuous functions with open domains, for which $f(x)$ tends to infinity as x approaches $\mathbf{bd}\ \mathbf{dom}\ f$.

Example – Quadratic minimization and least-squares

- The general convex quadratic minimization problem has the form

$$\text{minimize } (1/2)x^T Px + q^T x + r,$$

where $P \in \mathbf{S}_+^n$, $q \in \mathbf{R}^n$, and $r \in \mathbf{R}$.

- This problem can be solved via the optimality conditions,

$$Px^* + q = 0.$$

- When $P \succ 0$, there is a unique solution, $x^* = -P^{-1}q$.
- In the case when $P \notin \mathbf{S}_{++}$,
    - any solution of $Px^* = -q$ is optimal (if a solution exists);
    - if $Px^* = -q$ does not have a solution, then the problem is unbounded below.

Example – Unconstrained geometric programming

- As a second example, we consider an unconstrained geometric program in convex form,

$$\text{minimize } f(x) = \log \sum_{i=1}^{m} \exp(a_i^T x + b_i).$$

- The optimality condition is

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

  which in general has no analytical solution, so here we must resort to an iterative algorithm.

- Since **dom** $f = \mathbf{R}^n$ for this problem, any point can be chosen as the initial point $x^{(0)}$.

Analytic center of linear inequalities (1/2)

- We consider the optimization problem

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

where the domain of f is the open set

$$\textbf{dom } f = \left\{ x \mid a_i^T x < b_i, \quad i = 1, ..., m \right\}.$$

- The objective function f in this problem is called the **logarithmic barrier** for the inequalities $a_i^T x \leq b_i$.
- The solution of the problem, if it exists, is called the **analytic center** of the inequalities.

Analytic center of linear inequalities (2/2)

- The initial point $x^{(0)}$ must satisfy the strict inequalities

$$a_i^T x^{(0)} < b_i, i = 1, \ldots, m.$$

- Since f is closed, the sublevel set S for any such point is closed.

Strong convexity (1/2)

- We assume that the objective function is **strongly convex** on S: there exists an $m > 0$ such that

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

for all $x \in S$.

- If f is strongly convex, then for $x, y \in S$, there exists some z on the line segment $[x, y]$ such that

$$\begin{aligned} f(y) &= f(x) + \nabla f(x)^T (y-x) + \frac{1}{2}(y-x)^T \nabla^2 f(z)(y-x) \\ &\geq f(x) + \nabla f(x)^T (y-x) + \frac{m}{2}||y-x||_2^2. \end{aligned}$$

Strong convexity (2/2)

- When $m = 0$, we recover the basic inequality characterizing convexity; for $m > 0$ we obtain a better lower bound on $f(y)$ than follows from convexity alone.
- Note that $f(x) + \nabla f(x)^T(y - x) + \frac{m}{2}||y - x||_2^2$ can be minimized by $\tilde{y} = x - (1/m)\nabla f(x)$. Therefore we have

$$\begin{aligned} f(y) &\geq f(x) + \nabla f(x)^T(y - x) + \frac{m}{2}||y - x||_2^2 \\ &\geq f(x) + \nabla f(x)^T(\tilde{y} - x) + \frac{m}{2}||\tilde{y} - x||_2^2 \\ &= f(x) - \frac{1}{2m}||\nabla f(x)||_2^2. \end{aligned}$$

- Since this holds for any $y \in S$, we have

$$p^* \geq f(x) - \frac{1}{2m}||\nabla f(x)||_2^2.$$

Strong convexity and implications (1/2)

- This inequality shows that if the gradient $||\nabla f(x)||_2$ is small at some point x, then x is nearly optimal. Specifically,

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

- We can also derive a bound on $||x - x^*||_2$, the distance between x and any optimal point x^*, in terms of $||\nabla f(x)||_2$:

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

- One consequence of the above inequality is that the optimal point x^* is unique.

Strong convexity and implications (2/2)

- Proof of the inequality $||x - x^*||_2 \leq \frac{2}{m}||\nabla f(x)||_2$:

$$\begin{aligned} p^* = f(x^*) &\geq f(x) + \nabla f(x)^T(x^* - x) + \frac{m}{2}||x^* - x||_2^2 \\ &\geq f(x) - ||\nabla f(x)||_2 \, ||x^* - x||_2 + \frac{m}{2}||x^* - x||_2^2, \end{aligned}$$

where we use the Cauchy-Schwarz inequality in the second inequality. Since $p^* \leq f(x)$, we must have

$$-||\nabla f(x)||_2 \, ||x^* - x||_2 + \frac{m}{2}||x^* - x||_2^2 \leq 0. \text{ (QED)}$$

Upper bound on $\nabla^2 f(x)$ (1/2)

- The inequality

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{m}{2}||y - x||_2^2$$

  implies that the sublevel sets contained in S are bounded, so in particular, S is bounded.

- Therefore, the maximum eigenvalue of $\nabla^2 f(x)$, which is a continuous function of x on S, is bounded above on S, i.e., there exists a constant M such that

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

  for all $x \in S$.

Upper bound on $\nabla^2 f(x)$ (2/2)

- This upper bound on the Hessian implies for any $x, y \in S$,

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

- Minimizing each side over y yields

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

Condition number of convex sets (1/3)

- From the above discussions, we have

$$mI \preceq \nabla^2 f(x) \preceq MI$$

  for all $x \in S$.
- The ratio $\kappa = M/m$ is thus an upper bound on the **condition number** of the matrix $\nabla^2 f(x)$, i.e., the ratio of its largest eigenvalue to its smallest eigenvalue.
- We define the **width** of a convex set $C \subseteq \mathbf{R}^n$, in the direction q, where $||q||_2 = 1$, as

$$W(C, q) = \sup_{z \in C} q^T z - \inf_{z \in C} q^T z.$$

Condition number of convex sets (2/3)

- The **minimum width** and **maximum width** of C are given by

$$W_{min} = \inf_{||q||_2=1} W(C,q), W_{max} = \sup_{||q||_2=1} W(C,q).$$

- The **condition number** of the convex set C is defined as

$$\mathbf{cond}(C) = \frac{W_{max}^2}{W_{min}^2},$$

  i.e., the square of the ratio of its maximum width to its minimum width.

- The condition number of C gives a measure of its anisotropy or eccentricity.

Condition number of convex sets (3/3)

- If the condition number of a set C is small (say, near one) it means that the set has approximately the same width in all directions, i.e., it is nearly spherical.
- If the condition number is large, it means that the set is far wider in some directions than in others.

Example – Condition number of an ellipsoid (1/2)

- Let $\mathcal{E}$ be the ellipsoid

$$\mathcal{E} = \left\{ x \;\middle|\; (x - x_0)^T A^{-1} (x - x_0) \leq 1 \right\},$$

where $A \in \mathbf{S}^n_{++}$.

- The width of $\mathcal{E}$ in the direction q is

$$\begin{aligned} \sup_{z\in\mathcal{E}} q^T z - \inf_{z\in\mathcal{E}} q^T z &= (\|A^{1/2}q\|_2 + q^T x_0) - (-\|A^{1/2}q\|_2 + q^T x_0) \\ &= 2\|A^{1/2}q\|_2. \end{aligned}$$

Example – Condition number of an ellipsoid (2/2)

- So, the minimum and maximum width of $\mathcal{E}$ are

$$W_{min} = 2\lambda_{min}(A)^{1/2}, W_{max} = 2\lambda_{max}(A)^{1/2},$$

and the condition number is

$$\mathbf{cond}(\mathcal{E}) = \frac{\lambda_{max}(A)}{\lambda_{min}(A)} = \kappa(A),$$

where $\kappa(A)$ denotes the condition number of the matrix A, i.e., the ratio of its maximum singular value to its minimum singular value.

- Thus the condition number of the ellipsoid $\mathcal{E}$ is the same as the condition number of the matrix A that defines it.

Condition number of sublevel sets (1/3)

- Now suppose f satisfies $mI \preceq \nabla^2 f(x) \preceq MI$ for all $x \in S$.
- We will derive a bound on the condition number of the α-sublevel $C_\alpha = \{x \mid f(x) \leq \alpha\}$, where $p^* < \alpha \leq f(x^{(0)})$.
- Note that

$$p^* + (M/2)||y - x^*||_2^2 \geq f(y) \geq p^* + (m/2)||y - x^*||_2^2,$$

which implies that $B_{inner} \subseteq C_\alpha \subseteq B_{outer}$ where

$$B_{inner} = \left\{ y \mid ||y - x^*||_2 \leq (2(\alpha - p^*)/M)^{1/2} \right\},$$

$$B_{outer} = \left\{ y \mid ||y - x^*||_2 \leq (2(\alpha - p^*)/m)^{1/2} \right\}.$$

Condition number of sublevel sets (2/3)

- In other words, the α-sublevel set contains B_{inner}, and is contained in B_{outer}, which are balls with radii

$$(2(\alpha - p^*)/M)^{1/2}, (2(\alpha - p^*)/m)^{1/2},$$

respectively.

- The ratio of the radii squared gives an upper bound on the condition number of C_α:

$$\mathbf{cond}(C_\alpha) \leq \frac{M}{m}.$$

- We can also give a geometric interpretation of the condition number $\kappa(\nabla^2 f(x^*))$ of the Hessian at the optimum $\nabla^2 f(x^*)$.

Condition number of sublevel sets (3/3)

- From the Taylor series expansion of f around x^*,

$$f(y) \approx p^* + \frac{1}{2}(y - x^*)^T \nabla^2 f(x^*)(y - x^*),$$

we see that, for α close to p^*,

$$C_\alpha \approx \left\{ y \mid (y - x^*)^T \nabla^2 f(x^*)(y - x^*) \leq 2(\alpha - p^*) \right\},$$

i.e., the sublevel set is well approximated by an ellipsoid with center x^*.

- Therefore

$$\lim_{\alpha \to p^*} \mathbf{cond}(C_\alpha) = \kappa(\nabla^2 f(x^*)).$$

- We will see that the condition number of the sublevel sets of f (which is bounded by M/m) has a strong effect on the efficiency of common methods for unconstrained minimization.

Descent methods (1/3)

- The algorithms described in this lecture produce a minimizing sequence $x^{(k)}, \ k = 1, \dots,$ where

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

  and $t^{(k)} > 0$ (except when $x^{(k)}$ is optimal).
- The vector $\Delta x^{(k)} \in \mathbf{R}^n$ is called the step or search direction, and $k = 0, 1, \dots$ denotes the iteration number.
- The scalar $t^{(k)} \geq 0$ is called the step size or step length at iteration k (even though it is not equal to $||x^{(k+1)} - x^{(k)}||$ unless $||\Delta x^{(k)}|| = 1$).

Descent methods (2/3)

- When we focus on one iteration of an algorithm, we sometimes drop the superscripts and use the lighter notation $x^+ = x + t\Delta x$, or $x := x + t\Delta x$, in place of $x^{(k+1)} = x^{(k)} + t^{(k)}\Delta x^{(k)}$.
- All the methods we study are descent methods, which means that
$$f(x^{(k+1)}) < f(x^{(k)}),$$
except when $x^{(k)}$ is optimal.
- This implies that for all k we have $x^{(k)} \in S$, the initial sublevel set, and in particular we have $x^{(k)} \in \textbf{dom } f$.

Descent methods (3/3)

- From convexity we know that $\nabla f(x^{(k)})^T(y - x^{(k)}) \geq 0$ implies $f(y) \geq f(x^{(k)})$, so the search direction in a descent method must satisfy $\nabla f(x^{(k)})^T \Delta x^{(k)} < 0$, i.e., it must make an acute angle with the negative gradient.
- We call such a direction a **descent direction** (for f, at $x^{(k)}$).

General descent method (1/2)

- The outline of a general descent method is as follows, which alternates between two steps: determining a descent direction Δx, and the selection of a step size t.
- **Algorithm 1.** General descent method.
  given a starting point $x \in \textbf{dom } f$.
  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.

General descent method (2/2)

- The second step is called the **line search** (or ray search, to be more accurate) since selection of the step size t determines where along the line $\{x + t\Delta x \mid t \in \mathbf{R}^+\}$ the next iterate will be.
- A practical descent method has the same general structure, but might be organized differently.
  - For example, the stopping criterion is often checked while, or immediately after, the descent direction Δx is computed.
  - The stopping criterion is often of the form $||\nabla f(x)||_2 \leq \eta$, where η is small and positive, as suggested by the suboptimality condition

$$p^* \geq f(x) - \frac{1}{2m}||\nabla f(x)||_2^2.$$

Exact line search

- One line search method sometimes used in practice is **exact line search**, in which t is chosen to minimize f along the ray $\{x + t\Delta x \mid t \geq 0\}$:

$$t = \arg\min_{s \geq 0} f(x + s\Delta x).$$

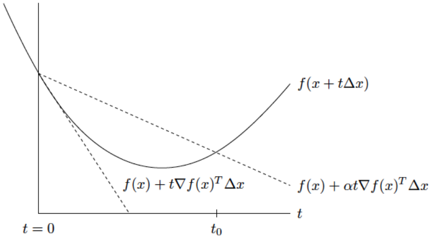
- An exact line search is used when the cost of the minimization problem with one variable is low compared to the cost of computing the search direction itself.

Backtracking line search (1/5)

- Most line searches used in practice are inexact: the step length is chosen to approximately minimize f along the ray $\{x + t\Delta x \mid t \geq 0\}$, or even to just reduce f 'enough'.
- Many inexact line search methods have been proposed. We study here one of them, called **backtracking line search**, which is very simple and quite effective.
- It depends on two constants α, β with $0 < \alpha < 0.5,\ 0 < \beta < 1$.

Backtracking line search (2/5)

- **Algorithm 2.** Backtracking line search.
  given a descent direction Δx for f at $x \in \textbf{dom } f$,
  $\alpha \in (0, 0.5)$, $\beta \in (0, 1)$.
  $t := 1$.
  while $f(x + t\Delta x) > f(x) + \alpha t \nabla f(x)^T \Delta x$,
  $t := \beta t$.



Backtracking line search (3/5)

- Since Δx is a descent direction, we have $\nabla f(x)^T \Delta x < 0$, so for small enough t we have

$$f(x + t\Delta x) \approx f(x) + t\nabla f(x)^T \Delta x < f(x) + \alpha t \nabla f(x)^T \Delta x,$$

  which shows that the backtracking line search eventually terminates.
- The constant α can be interpreted as the fraction of the decrease in f predicted by linear extrapolation that we will accept.
- This figure suggests, and it can be shown, that the backtracking exit inequality $f(x + t\Delta x) \leq f(x) + \alpha t \nabla f(x)^T \Delta x$ holds for $t \geq 0$ in an interval $(0, t_0]$.

Backtracking line search (4/5)

- It follows that the backtracking line search stops with a step length t that satisfies $t = 1$, or $t \in (\beta t_0, t_0]$.
- The first case occurs when the step length $t = 1$ satisfies the backtracking condition, i.e., $1 \leq t_0$.
- In particular, we can say that the step length obtained by backtracking line search satisfies

$$t \geq \min\{1, \beta t_0\}.$$

- When **dom** f is not all of $\mathbf{R}^n$, the condition $f(x + t\Delta x) \leq f(x) + \alpha t \nabla f(x)^T \Delta x$ in the backtracking line search must be interpreted carefully.
- By our convention that f is infinite outside its domain, the inequality implies that $x + t\Delta x \in$ **dom** f.

Backtracking line search (5/5)

- In a practical implementation, we first multiply t by β until $x + t\Delta x \in \textbf{dom } f$; then we start to check whether the inequality

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

  holds.
- The parameter α is typically chosen between 0.01 and 0.3, meaning that we accept a decrease in f between 1% and 30% of the prediction based on the linear extrapolation.
- The parameter β is often chosen to be between 0.1 (which corresponds to a very crude search) and 0.8 (which corresponds to a less crude search).

Gradient descent method

- A natural choice for the search direction is the negative gradient $\Delta x = -\nabla f(x)$. The resulting algorithm is called the **gradient algorithm** or **gradient descent method**.
- **Algorithm 3.** Gradient descent method.
  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.
- The stopping criterion is usually of the form $||\nabla f(x)||_2 \leq \eta$, where η is small and positive.

Example – A quadratic problem in $\mathbf{R}^2$ (1/3)

- We first consider a simple example with the quadratic objective function on $\mathbf{R}^2$

$$f(x) = \frac{1}{2}(x_1^2 + \gamma x_2^2),$$

  where $\gamma > 0$.
- Clearly, the optimal point is $x^* = 0$, and the optimal value is 0.
- The Hessian of f is constant, and has eigenvalues 1 and γ, so the condition numbers of the sublevel sets of f are all exactly

$$\frac{\max\{1, \gamma\}}{\min\{1, \gamma\}} = \max\{\gamma, 1/\gamma\}.$$

Example – A quadratic problem in $\mathbf{R}^2$ (2/3)

- The tightest choices for the strong convexity constants m and M are

$$m = \min\{1, \gamma\}, M = \max\{1, \gamma\}.$$

- We apply the gradient descent method with exact line search, starting at the point $x^{(0)} = (\gamma, 1)$.
- It can be shown that the kth iterate $x^{(k)}$ has the closed-form expression as follows:

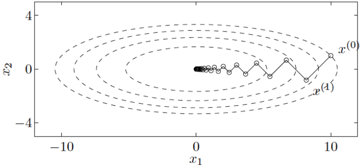
$$x_1^{(k)} = \gamma\left(\frac{\gamma - 1}{\gamma + 1}\right)^k, \quad x_2^{(k)} = \left(-\frac{\gamma - 1}{\gamma + 1}\right)^k,$$

and the corresponding function value is

$$f(x^{(k)}) = \frac{\gamma(\gamma + 1)}{2}\left(\frac{\gamma - 1}{\gamma + 1}\right)^{2k} = \left(\frac{\gamma - 1}{\gamma + 1}\right)^{2k} f(x^{(0)}).$$

Example – A quadratic problem in $\mathbf{R}^2$ (3/3)

- This case for $\gamma = 10$ is illustrated below.



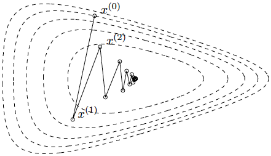
- For this simple example, convergence is exactly linear, i.e., the error is exactly a geometric series, reduced by the factor $|(\gamma - 1)/(\gamma + 1)|^2$ at each iteration.
- For $\gamma = 1$, the exact solution is found in one iteration; for γ not far from one (say, between 1/3 and 3) convergence is rapid.
- The convergence is very slow for $\gamma \gg 1$ or $\gamma \ll 1$.

Example – A nonquadratic problem in $\mathbf{R}^2$ (1/4)

- We now consider a nonquadratic example in $\mathbf{R}^2$, with

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

- We apply the gradient method with a backtracking line search, with $\alpha = 0.1, \beta = 0.7$.
- The following figure shows some level curves of f, and the iterates $x^{(k)}$ generated by the gradient method (shown as small circles).

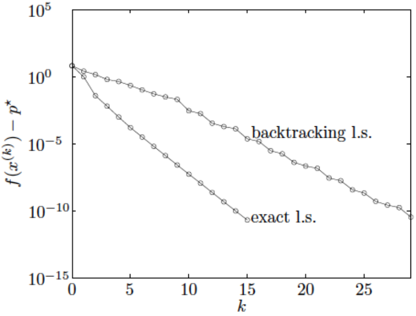


Example – A nonquadratic problem in $\mathbf{R}^2$ (2/4)

- The lines connecting successive iterates show the scaled steps,

$$x^{(k+1)} - x^{(k)} = -t^{(k)} \nabla f(x^{(k)}).$$

- The figure below shows the error $f(x^{(k)}) - p^*$ versus iteration k.

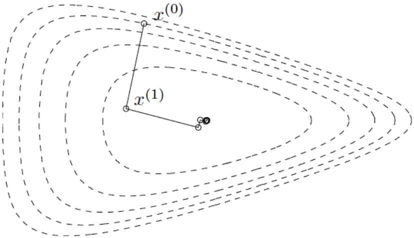


Example – A nonquadratic problem in $\mathbf{R}^2$ (3/4)

- The plot reveals that the error converges to zero approximately as a geometric series.
- In this example, the error is reduced from about 10 to about 10^{-7} in 20 iterations, so the error is reduced by a factor of approximately $10^{-8/20} \approx 0.4$ each iteration.
- This reasonably rapid convergence is predicted by our convergence analysis, since the sublevel sets of f are not too badly conditioned, which in turn means that M/m can be chosen as not too large.
- To compare backtracking line search with an exact line search, we use the gradient method with an exact line search, on the same problem, and with the same starting point.

Example – A nonquadratic problem in $\mathbf{R}^2$ (4/4)

- The results are given in the following figure. Here too the convergence is approximately linear, about twice as fast as the gradient method with backtracking line search.



- With exact line search, the error is reduced by about 10^{-11} in 15 iterations, i.e., a reduction by a factor of about $10^{-11/15} \approx 0.2$ per iteration.

Gradient method and condition number (1/3)

- Our last experiment will illustrate the importance of the condition number of $\nabla^2 f(x)$ (or the sublevel sets) on the rate of convergence of the gradient method.
- We start with the function given by

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

but replace the variable x by $x = T\bar{x}$, where

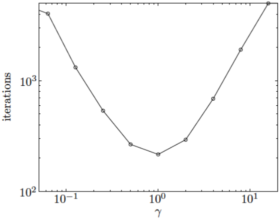
$$T = \textbf{diag}\left((1, \gamma^{1/n}, \gamma^{2/n}, ..., \gamma^{(n-1)/n})\right),$$

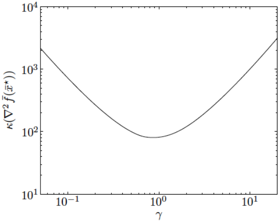
i.e., we minimize

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

Gradient method and condition number (2/3)

- This gives us a family of optimization problems, indexed by γ, which affects the problem condition number. We use a backtracking line search with $\alpha = 0.3$ and $\beta = 0.7$.
- The left figure shows the number of iterations required to achieve $\bar{f}(\bar{x}^{(k)}) - \bar{p}^* < 10^{-5}$ as a function of γ, and the condition number of the Hessian $\nabla^2 \bar{f}(\bar{x}^*)$ versus γ at the optimum is shown on the right.





Gradient method and condition number (3/3)

- For large and small γ, the condition number increases roughly as $\max\{\gamma^2, 1/\gamma^2\}$, in a very similar way as the number of iterations depends on γ.
- This shows again that the relation between conditioning and convergence speed is a real phenomenon, and not just an artifact of our analysis.

Summary for Gradient Descent

- From the numerical results shown, we make the following summary.
  - The gradient method often exhibits approximately linear convergence, i.e., the error $f(x^{(k)}) - p^*$ converges to zero approximately as a geometric series.
  - The choice of backtracking parameters α, β has a noticeable but not dramatic effect on the convergence.
  - An exact line search sometimes improves the convergence of the gradient method, but the effect is not large.
  - The convergence rate depends greatly on the condition number of the Hessian, or the sublevel sets. When the condition number is large (say, 1000 or more) the gradient method is so slow that it is useless in practice.
- The main advantage of the gradient method is its simplicity.
- Its main disadvantage is that its convergence rate depends so critically on the condition number of the Hessian or sublevel sets.

Steepest descent method (1/2)

- The first-order Taylor approximation of $f(x+v)$ around x is

$$f(x+v) \approx \hat{f}(x+v) = f(x) + \nabla f(x)^T v,$$

  where the term $\nabla f(x)^T v$ is the **directional derivative** of f at x in the direction v.
- It gives the approximate change in f for a small step v.
- The step v is a descent direction if the directional derivative is negative.

Steepest descent method (2/2)

- Let $||\cdot||$ be any norm on $\mathbf{R}^n$. We define a **normalized steepest descent direction** (with respect to the norm $||\cdot||$) as

$$\Delta x_{\mathrm{nsd}} = \arg\min \left\{ \nabla f(x)^T v \;\middle|\; ||v|| = 1 \right\}.$$

  which is a step of unit norm that gives the largest decrease in the linear approximation of f.
- It is convenient to consider a **steepest descent step** Δx_{sd} that is unnormalized, by scaling the normalized steepest descent direction in a particular way:

$$\Delta x_{\mathrm{sd}} = ||\nabla f(x)||_* \Delta x_{\mathrm{nsd}}.$$

- Note that for the steepest descent step, we have

$$\nabla f(x)^T \Delta x_{\mathrm{sd}} = ||\nabla f(x)||_* \nabla f(x)^T \Delta x_{\mathrm{nsd}} = -||\nabla f(x)||_*^2$$

Steepest descent Algorithm

- The steepest descent method uses the steepest descent direction as search direction.
- **Algorithm 4.** Steepest descent method.
  given a starting point $x \in$ **dom** f.
  repeat
  1. Compute steepest descent direction Δx_{sd}.
  2. *Line search*. Choose t via backtracking or exact line search.
  3. *Update*. $x := x + t\Delta x_{\mathrm{sd}}$.

  until stopping criterion is satisfied.
- When exact line search is used, scale factors in the descent direction have no effect, so the normalized or unnormalized direction can be used.

Steepest descent for Euclidean norm

- If we take the norm $||\cdot||$ to be the Euclidean norm, then the steepest descent direction is simply the negative gradient, i.e., $\Delta x_{\mathrm{sd}} = -\nabla f(x)$.
- The steepest descent method for the Euclidean norm coincides with the gradient descent method.

Steepest descent for quadratic norm (1/2)

- We consider the quadratic norm

$$||z||_P = (z^T Pz)^{1/2} = ||P^{1/2}z||_2,$$

where $P \in \mathbf{S}^n_{++}$.

- The normalized steepest descent direction is given by

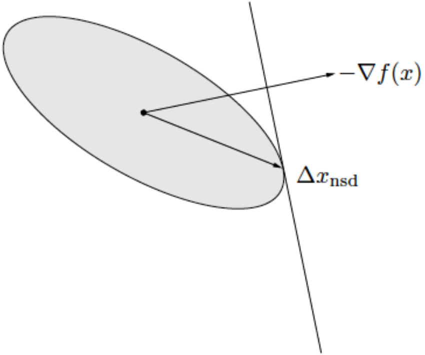
$$\Delta x_{\text{nsd}} = -\left(\nabla f(x)^T P^{-1} \nabla f(x)\right)^{-1/2} P^{-1} \nabla f(x).$$

- The dual norm is given by $||z||_* = ||P^{-1/2}z||_2$, so the steepest descent step with respect to $||\cdot||_P$ is given by

$$\Delta x_{\text{sd}} = -P^{-1} \nabla f(x).$$

Steepest descent for quadratic norm (2/2)

- The normalized steepest descent direction for a quadratic norm is illustrated in the following figure.



Interpretation via change of coordinates (1/2)

- The steepest descent direction Δx_{sd} can be interpreted as the gradient search direction after a change of coordinates is applied to the problem.
- Define $\bar{u} = P^{1/2}u$, so we have $||u||_P = ||\bar{u}||_2$. Using this change of coordinates, we can solve the original problem of minimizing f by solving the equivalent problem of minimizing the function $\bar{f} : \mathbf{R}^n \to \mathbf{R}$, given by

$$\bar{f}(\bar{u}) = f(P^{-1/2}\bar{u}) = f(u).$$

- If we apply the gradient method to $\bar{f}$, the search direction at a point $\bar{x}$ (corresponding to $x = P^{-1/2}\bar{x}$ for the original problem) is

$$\Delta\bar{x} = -\nabla\bar{f}(\bar{x}) = -P^{-1/2}\nabla f(P^{-1/2}\bar{x}) = -P^{-1/2}\nabla f(x).$$

Interpretation via change of coordinates (2/2)

- This gradient search direction corresponds to the direction

$$\Delta x = P^{-1/2}(-P^{-1/2}\nabla f(x)) = -P^{-1}\nabla f(x)$$

  for the original variable x.
- In other words, the steepest descent method in the quadratic norm $||\cdot||_P$ can be thought of as the gradient method applied to the problem after the change of coordinates $\bar{x} = P^{1/2}x$.

Steepest descent for ℓ_1-norm (1/3)

- We consider the steepest descent method for the ℓ_1-norm.
- A normalized steepest descent direction w.r.t. ℓ_1-norm is

$$\Delta x_{\text{nsd}} = \arg\min \left\{ \nabla f(x)^T v \mid ||v||_1 \leq 1 \right\}.$$

- Let i be any index for which $||\nabla f(x)||_\infty = |(\nabla f(x))_i|$.
- Then a normalized steepest descent direction Δx_{nsd} for the ℓ_1-norm is given by

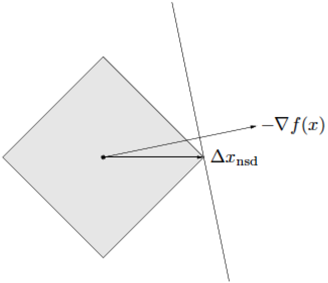
$$\Delta x_{\text{nsd}} = -\textbf{sign}\left(\frac{\partial f(x)}{\partial x_i}\right) e_i,$$

where e_i is the ith standard basis vector.

Steepest descent for ℓ_1-norm (2/3)

- An unnormalized steepest descent step is then

$$\Delta x_{\mathrm{sd}} = \Delta x_{\mathrm{nsd}} ||\nabla f(x)||_\infty = -\frac{\partial f(x)}{\partial x_i} e_i.$$



Steepest descent for ℓ_1-norm (3/3)

- The steepest descent algorithm in the ℓ_1-norm has a very natural interpretation: At each iteration we select a component of $\nabla f(x)$ with maximum absolute value, and then decrease or increase the corresponding component of x, according to the sign of $(\nabla f(x))_i$.
- The algorithm is sometimes called a coordinate-descent algorithm, since only one component of the variable x is updated at each iteration. This can greatly simplify, or even trivialize, the line search.

Choice of norm for steepest descent (1/2)

- The choice of norm used to define the steepest descent direction can have a dramatic effect on the convergence rate.
- We consider the case of steepest descent with quadratic P-norm.
- Recall that the steepest descent method with quadratic P-norm is the same as the gradient method applied to the problem after the change of coordinates $\bar{x} = P^{1/2}x$.
- We know that the gradient method works well when the condition numbers of the sublevel sets (or the Hessian near the optimal point) are moderate, and works poorly when the condition numbers are large.

Choice of norm for steepest descent (2/2)

- So, when the sublevel sets, after the change of coordinates $\bar{x} = P^{1/2}x$, are moderately conditioned, the steepest descent method will work well.
- This observation provides a prescription for choosing P. For example, if an approximation $\hat{H}$ of the Hessian at the optimal point $H(x^*)$ were known, a very good choice of P would be $P = \hat{H}$, since the Hessian of $\tilde{f}$ at the optimum is then

$$\hat{H}^{-1/2}\nabla^2 f(x^*)\hat{H}^{-1/2} \approx I,$$

and so is likely to have a low condition number.

Examples (1/5)

- We illustrate some of these ideas using the nonquadratic problem in $\mathbf{R}^2$ with objective function

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

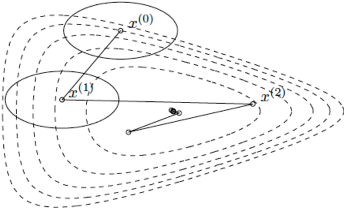
- We apply the steepest descent method to the problem, using the two quadratic norms defined by

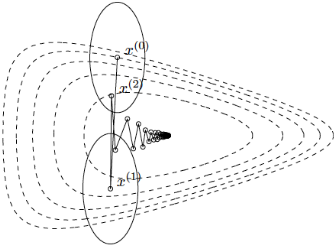
$$P_1 = \begin{bmatrix} 2 & 0 \\ 0 & 8 \end{bmatrix}, P_2 = \begin{bmatrix} 8 & 0 \\ 0 & 2 \end{bmatrix}.$$

- In both cases we use a backtracking line search with $\alpha = 0.1$ and $\beta = 0.7$.

Examples (2/5)

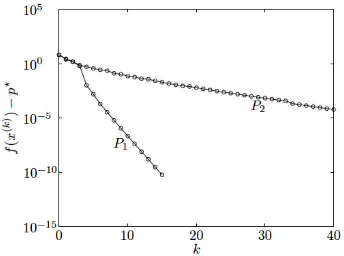
- The following figures show the iterates for steepest descent with norm $||\cdot||_{P_1}$ and norm $||\cdot||_{P_2}$, respectively.





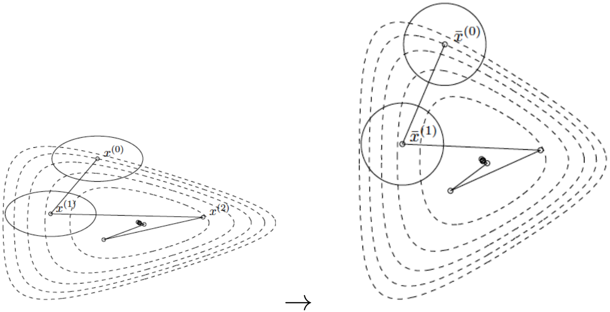
Examples (3/5)

- The following figure shows the error versus iteration number for both norms and shows that the choice of norm strongly influences the convergence.
- With the norm $||\cdot||_{P_1}$, convergence is a bit more rapid than the gradient method, whereas with the norm $||\cdot||_{P_2}$, convergence is far slower.



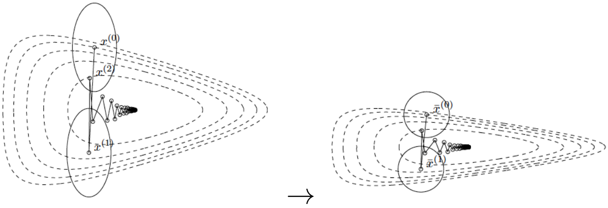
Examples (4/5)

- This can be explained by examining the problems after the changes of coordinates $\bar{x} = P_1^{1/2}x$ and $\bar{x} = P_2^{1/2}x$, respectively.
- The change of variables associated with P_1 yields sublevel sets with modest condition number, so convergence is fast.



Examples (5/5)

- The change of variables associated with P_2 yields sublevel sets that are more poorly conditioned, which explains the slower convergence.



The Newton step

Newton step

For $x \in \textbf{dom } f$, the vector

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

is called the Newton step (for f, at x).

- If $\nabla^2 f(x)$ is positive definite, it implies that

$$\nabla f(x)^T \Delta x_{\mathrm{nt}} = -\nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x) < 0$$

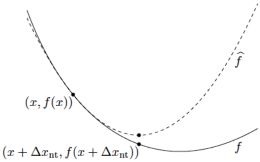
  unless $\nabla f(x) = 0$, so the Newton step is a descent direction (unless x is optimal).
- The Newton step can be interpreted and motivated in several ways.

Minimizer of second-order approximation (1/2)

- The second-order Taylor approximation $\hat{f}$ of f at x is

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

  which is a convex quadratic function of v, and is minimized when $v = \Delta x_{\mathrm{nt}}$.
- Thus, the Newton step Δx_{nt} is what should be added to the point x to minimize the second-order approximation of f at x.



Minimizer of second-order approximation (2/2)

- If the function f is quadratic, then $x + \Delta x_{\mathrm{nt}}$ is the exact minimizer of f.
- If the function f is nearly quadratic, intuition suggests that $x + \Delta x_{\mathrm{nt}}$ should be a very good estimate of the minimizer of f, i.e., x^*.
- Since f is twice differentiable, the quadratic model of f will be very accurate when x is near x^*. It follows that when x is near x^*, the point $x + \Delta x_{\mathrm{nt}}$ should be a very good estimate of x^*.

Steepest descent direction in Hessian norm (1/2)

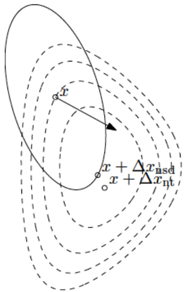
- The Newton step is also the steepest descent direction at x, for the quadratic norm defined by the Hessian $\nabla^2 f(x)$, i.e.,

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

- This gives another insight into why the Newton step should be a good search direction, and a very good search direction when x is near x^*.
- Recall that steepest descent, with quadratic norm $|| \cdot ||_P$, converges very rapidly when the Hessian, after the associated change of coordinates, has small condition number.
- In particular, near x^*, a very good choice is $P = \nabla^2 f(x^*)$.

Steepest descent direction in Hessian norm (2/2)

- When x is near x^*, we have $\nabla^2 f(x) \approx \nabla^2 f(x^*)$, which explains why the Newton step is a very good choice of search direction.



- In the above figure, the arrow denotes the gradient descent direction.

Solution of linearized optimality condition (1/2)

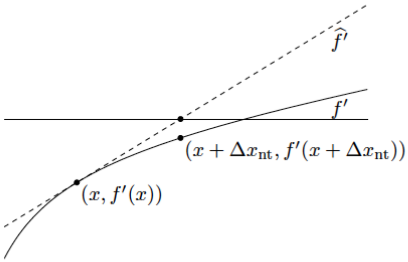
- We can linearize the optimality condition $\nabla f(x^*) = 0$ near x and obtain

$$\nabla f(x + v) \approx \nabla f(x) + \nabla^2 f(x) v = 0,$$

  which is a linear equation in v, with solution $v = \Delta x_{\mathrm{nt}}$.
- So the Newton step Δx_{nt} is what must be added to x so that the linearized optimality condition holds.
- This suggests that when x is near x^* (so the optimality conditions almost hold), the update $x + \Delta x_{\mathrm{nt}}$ should be a very good approximation of x^*.
- When $n = 1$, i.e., $f : \mathbf{R} \to \mathbf{R}$, this interpretation is particularly simple.

Solution of linearized optimality condition (2/2)

- The solution x^* of the minimization problem is characterized by $f'(x^*) = 0$, i.e., it is the zero-crossing of the derivative f', which is monotonically increasing since f is convex.
- Given our current approximation x of the solution, we form a first-order Taylor approximation of f' at x.
- The zero-crossing of this affine approximation is then $x + \Delta x_{\mathrm{nt}}$.



Affine invariance of the Newton step

- An important feature of the Newton step is that it is independent of linear (or affine) changes of coordinates.
- Suppose $T \in \mathbf{R}^{n\times n}$ is nonsingular, and define $\bar{f}(y) = f(Ty)$. Then we have $\nabla \bar{f}(y) = T^T \nabla f(x), \nabla^2 \bar{f}(y) = T^T \nabla^2 f(x) T$, where $x = Ty$.
- The Newton step for $\bar{f}$ at y is therefore

$$\begin{aligned}\Delta y_{nt} &= -(T^T \nabla^2 f(x) T)^{-1}(T^T \nabla f(x)) \\ &= -T^{-1} \nabla^2 f(x)^{-1} \nabla f(x) \\ &= T^{-1}\Delta x_{\mathrm{nt}},\end{aligned}$$

where Δx_{nt} is the Newton step for f at x. Hence the Newton steps of f and $\bar{f}$ are related by the same linear transformation.

$$x + \Delta x_{\mathrm{nt}} = T(y + \Delta y_{nt}).$$

The Newton decrement (1/2)

- The quantity

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

  is called the **Newton decrement** at x.

- We can relate the Newton decrement to the quantity

$$f(x) - \inf_y \hat{f}(y),$$

  where $\hat{f}$ is the second-order approximation of f at x:

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

The Newton decrement (2/2)

- Thus, $\lambda^2/2$ is an estimate of $f(x) - p^*$, based on the quadratic approximation of f at x.
- We can also express the Newton decrement as

$$\lambda(x) = \left(\Delta x_{nt}^T \nabla^2 f(x) \Delta x_{\mathrm{nt}}\right)^{1/2},$$

which shows that λ is the norm of the Newton step, in the quadratic norm defined by the Hessian, i.e., the norm

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

- The Newton decrement is, like the Newton step, affine invariant: the Newton decrement of $\bar{f}(y) = f(Ty)$ at y, where T is nonsingular, is the same as the Newton decrement of f at $x = Ty$.

Newton's method

- Newton's method, as outlined below, is sometimes called the **damped Newton method**, to distinguish it from the pure Newton method, which uses a fixed step size $t = 1$.
- **Algorithm 5.** (Damped) Newton's method.
  given a starting point $x \in \mathbf{dom}\ f$, tolerance $\epsilon > 0$.
  repeat
  1. *Compute the Newton step and decrement.*
  $$\Delta x_{\mathrm{nt}} := -\nabla^2 f(x)^{-1} \nabla f(x); \lambda^2 := \nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x).$$
  2. *Stopping criterion.* **quit** if $\lambda^2/2 \leq \epsilon$.
  3. *Line search.* Choose step size t by backtracking line search.
  4. *Update.* $x := x + t\Delta x_{\mathrm{nt}}$.
- This is essentially the general descent method using the Newton step as search direction.

Convergence analysis (1/3)

- We assume, as before, that f is twice continuously differentiable, and strongly convex with constant m, i.e., $\nabla^2 f(x) \succeq mI$ for $x \in S$. This implies that there exists an $M > 0$ such that $\nabla^2 f(x) \preceq MI$ for all $x \in S$.
- In addition, we assume that the Hessian of f is **Lipschitz continuous** on S with constant L, i.e.,

$$||\nabla^2 f(x) - \nabla^2 f(y)||_2 \leq L||x - y||_2$$

  for all $x, y \in S$.
- The coefficient L, which can be interpreted as a bound on the third derivative of f, can be taken to be zero for a quadratic function.

Convergence analysis (2/3)

- More generally L measures how well f can be approximated by a quadratic model, so we can expect the **Lipschitz constant** L to play a critical role in the performance of Newton's method.
- Intuition suggests that Newton's method will work very well for a function whose quadratic model varies slowly (i.e., has small L).
- It can be shown that there are numbers η and γ with $0 < \eta \leq m^2/L$ and $\gamma > 0$ such that the following hold.
  - If $||\nabla f(x^{(k)})||_2 \geq \eta$, then

$$f(x^{(k+1)}) - f(x^{(k)}) \leq -\gamma.$$

Convergence analysis (3/3)

- If $||\nabla f(x^{(k)})||_2 < \eta$, then the backtracking line search selects $t^{(k)} = 1$ and

$$\frac{L}{2m^2}||\nabla f(x^{(k+1)})||_2 \leq \left(\frac{L}{2m^2}||\nabla f(x^{(k)})||_2\right)^2.$$

- The case when $||\nabla f(x^{(k)})||_2 \geq \eta$ is referred to as the **damped Newton phase**; the case $||\nabla f(x^{(k)})||_2 < \eta$ is called the **quadratically convergence phase**.
- The number of iterations needed is bounded above by

$$6 + \frac{M^2 L^2 / m^5}{\alpha\beta \min\{1, 9(1-2\alpha)^2\}}(f(x^{(0)}) - p^*).$$

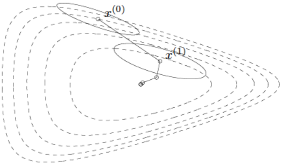
- The proof is omitted here. Interested audience can refer to the textbook.

Example in $\mathbf{R}^2$ (1/3)

- We apply Newton's method with backtracking line search, with parameters $\alpha = 0.1, \beta = 0.7$, on the test function $f(x_1, x_2) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1}$.
- The next figure shows the Newton iterates and the ellipsoids

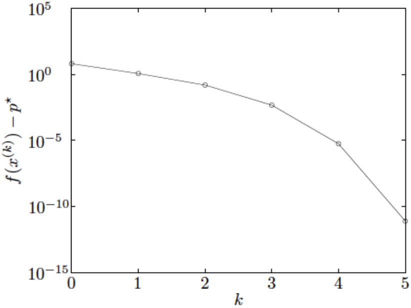
$$\left\{x \mid ||x - x^{(k)}||_{\nabla^2 f(x^{(k)})} \leq 1\right\}$$

for the first two iterates $k = 0, 1$.



Example in $\mathbf{R}^2$ (2/3)

- The method works well because these ellipsoids give good approximations of the shape of the sublevel sets.
- The error versus iteration number for the same example is shown below



Example in $\mathbf{R}^2$ (3/3)

- This plot shows that convergence to a very high accuracy is achieved in only five iterations.
- Quadratic convergence is clearly apparent: The last step reduces the error from about 10^{-5} to 10^{-10}.

Summary (1/2)

- Newton's method has several very strong advantages over gradient and steepest descent methods:
  - Convergence of Newton's method is rapid in general, and quadratic near x^*. Once the quadratic convergence phase is reached, at most six or so iterations are required to produce a solution of very high accuracy.
  - Newton's method is affine invariant.
  - It is insensitive to the choice of coordinates, or the condition number of the sublevel sets of the objective.
  - Newton's method scales well with problem size.
  - Its performance on problems in $\mathbf{R}^{10000}$ is similar to its performance on problems in $\mathbf{R}^{10}$, with only a modest increase in the number of steps required.
  - The good performance of Newton's method is not dependent on the choice of algorithm parameters.

Summary (2/2)

- In contrast, the choice of norm for steepest descent plays a critical role in its performance.
- The main disadvantage of Newton's method is the cost of forming and storing the Hessian, and the cost of computing the Newton step, which requires solving a set of linear equations.