AMSC 607 / CMSC 764 Advanced Numerical Optimization

Fall 2010

UNIT 2: Unconstrained Optimization, Part 1

Dianne P. O'Leary

Unconstrained Optimization: Fundamentals

The plan:

- How do we recognize a solution?
- Some geometry.
- Our basic algorithm for finding a solution.
- The model method: Newton.
- How close to Newton do we need to be?
- Making methods safe:
  - Descent directions and line searches.
  - Trust regions.

Note:

-
- References to books are supplementary and optional.

References for this set of notes: N&S Chapter 2 and Chapter 10.

How do we recognize a solution?

How do we recognize a solution?

Problem P: Given a function $f : S \to \mathcal{R}$, find

$$\min_{\mathbf{x} \in S} f(\mathbf{x})$$

with solution $\mathbf{x}^*$.

The point $\mathbf{x}^*$ is called the minimizer, and the value $f(\mathbf{x}^*)$ is the minimum.

For unconstrained optimization, the set S is usually taken to be $\mathcal{R}^n$, but sometimes we make use of upper or lower bounds on the variables, restricting our search to a box

$$\{\mathbf{x} : \boldsymbol{\ell} \leq \mathbf{x} \leq \mathbf{u}\}$$

for some given vectors $\boldsymbol{\ell}, \mathbf{u} \in \mathcal{R}^n$.

What does it mean to be a solution?

The point $\mathbf{x}^*$ is a local solution to Problem P if there is a $\delta > 0$ so that if $\mathbf{x} \in S$ and $\|\mathbf{x} - \mathbf{x}^*\| < \delta$, then $f(\mathbf{x}^*) \leq f(\mathbf{x})$.

In other words, $\mathbf{x}^*$ is at least as good as any point in its neighborhood.

The point $\mathbf{x}^*$ is a global solution to Problem P if for any $\mathbf{x} \in S$, then $f(\mathbf{x}^*) \leq f(\mathbf{x})$.

Note: It would be nice if every local solution was guaranteed to be global. This is true if f is convex. We'll look at this case more carefully in the "Geometry" section of these notes.

Some notation

We'll assume throughout the course that f is smooth enough that it has as many continuous derivatives as we need. For this section, that means 2 continuous derivatives plus one more, possibly discontinuous.

The gradient of f at $\mathbf{x}$ is defined to be the vector

$$\mathbf{g}(\mathbf{x}) = \bigtriangledown f(\mathbf{x}) = \begin{bmatrix} \partial f/\partial x_1 \\ \vdots \\ \partial f/\partial x_n \end{bmatrix}.$$

The Hessian of f at $\mathbf{x}$ is the derivative of the gradient:

$$\mathbf{H}(\mathbf{x}) = \bigtriangledown^2 f(\mathbf{x}), \text{ with } h_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$

Note that the Hessian is symmetric, unless f fails to be smooth enough.

How do we recognize a solution?

Recall from calculus Taylor series: Suppose we have a vector $\mathbf{p} \in \mathcal{R}^n$ with $\|\mathbf{p}\| = 1$, and a small scalar h. Then

$$f(\mathbf{x}^* + h\mathbf{p}) = f(\mathbf{x}^*) + h\mathbf{p}^T\mathbf{g}(x^*) + \frac{1}{2}h^2\mathbf{p}^T\mathbf{H}(\mathbf{x}^*)\mathbf{p} + O(h^3)\,.$$

First Order Necessary Condition for Optimality

$$f(\mathbf{x}^* + h\mathbf{p}) = f(\mathbf{x}^*) + h\mathbf{p}^T\mathbf{g}(\mathbf{x}^*) + \frac{1}{2}h^2\mathbf{p}^T\mathbf{H}(\mathbf{x}^*)\mathbf{p} + O(h^3)\,.$$

Now suppose that $\mathbf{g}(\mathbf{x}^*)$ is nonzero. Then we can always find a descent or downhill direction $\mathbf{p}$ so that

$$\mathbf{p}^T\mathbf{g}(\mathbf{x}^*) < 0\,.$$

(Take, for example, $\mathbf{p} = -\mathbf{g}(\mathbf{x}^*)/\|\mathbf{g}(\mathbf{x}^*)\|$.)

Therefore, for small enough h, we can make $\frac{1}{2}h^2\mathbf{p}^T\mathbf{H}(\mathbf{x}^*)\mathbf{p}$ small enough that

$$f(\mathbf{x}^* + h\mathbf{p}) < f(\mathbf{x}^*).$$

Therefore, a necessary condition for $\mathbf{x}^*$ to be a minimizer is that $\mathbf{g}(\mathbf{x}^*) = \mathbf{0}$.

Second Order Necessary Condition for Optimality

So we know that if $\mathbf{x}^*$ is a minimizer, then $\mathbf{g}(\mathbf{x}^*) = \mathbf{0}$, so

$$f(\mathbf{x}^* + h\mathbf{p}) = f(\mathbf{x}^*) + \frac{1}{2}h^2\mathbf{p}^T\mathbf{H}(\mathbf{x}^*)\mathbf{p} + O(h^3)\,.$$

Now suppose that we had a direction $\mathbf{p}$ so that $\mathbf{p}^T\mathbf{H}(\mathbf{x}^*)\mathbf{p} < 0$. (We call this a direction of negative curvature.) Then again, for small enough h, we could make $f(\mathbf{x}^* + h\mathbf{p}) < f(\mathbf{x}^*)$.

Therefore, a necessary condition for $\mathbf{x}^*$ to be a minimizer is that there be no direction of negative curvature.

From linear algebra, this is equivalent to saying that the matrix $\mathbf{H}(\mathbf{x}^*)$ must be positive semidefinite. In other words, all of its eigenvalues must be nonnegative.

Are these conditions sufficient?

Not quite.

Example: Let f be a function of a single variable:

$$f(x) = x^3.$$

Then $f'(x) = 3x^2$ and $f''(x) = 6x$, so $f'(0) = 0$ and $f''(0) = 0$, so $x = 0$ satisfies the first- and second-order necessary conditions for optimality, but it is not a minimizer of f. []

We are very close to sufficiency, though: Recall that a symmetric matrix is positive definite if all of its eigenvalues are positive.

If $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ and $\mathbf{H}(\mathbf{x})$ is positive definite, then $\mathbf{x}$ is a local minimizer.

Unquiz: Prove this. []

Some geometry

What all of this means geometrically

Imagine you are at point $\mathbf{x}$ on a mountain, described by the function $f(\mathbf{x})$, and it is foggy. (So $\mathbf{x} \in \mathcal{R}^2$.)

The direction $\mathbf{g}(\mathbf{x})$ is the direction of steepest ascent. So if you want to climb the mountain, it is the best direction to walk.

The direction $-\mathbf{g}(\mathbf{x})$ is the direction of steepest descent, the fastest way down.

Any direction $\mathbf{p}$ that makes a positive inner product with the gradient is an uphill direction, and any direction that makes a negative inner product is downhill.

If you are standing at a point where the gradient is zero, then there is no ascent direction and no descent direction, but a direction of positive curvature will lead you to a point where you can go uphill, and a direction of negative curvature will lead you to a point where you can descend.

If you can't find any of these, then you are at the bottom of a valley!

Geometry of sets

Unquiz: Let the set S be defined by

$$S = \{\mathbf{x} \in \mathcal{R}^n : \mathbf{c}(\mathbf{x}) \geq \mathbf{0}\}.$$

Draw S for each of these examples:

1.

$$\begin{aligned} -x_1 - x_2 &\geq -1 \\ x_1 &\geq 0 \\ x_2 &\geq 0 \end{aligned}$$

Notational note: $-x_1 - x_2 \geq -1$ means $c_1(x) = -x_1 - x_2 + 1$.

2.

$$\begin{aligned} -x_1 - x_2 &\geq -1 \\ x_1 + x_2 &\geq 1 \end{aligned}$$

3.

$$\begin{aligned} -x_1^2 - x_2^2 &\geq -1 \\ x_1 &\geq 0 \\ x_2 &\geq 0 \end{aligned}$$

4.

$$\begin{aligned} x_1^2 + x_2^2 &\geq 1 \\ x_1 &\geq 0 \\ x_2 &\geq 0 \end{aligned}$$

[]

Unquiz: Show that

$$c_j(x) = 0\,, \;\; j = 1, \ldots, m$$

if and only if

$$c_j(x) \geq 0\,, \;\; j = 1, \ldots, m,$$
$$c_{m+1}(x) = -c_1(x) - \ldots - c_m(x) \geq 0\,.$$

[]

Some jargon about sets

Sets may be

- bounded or unbounded. A set is bounded if we can draw a finite (hyper)sphere around it.
- convex or nonconvex. A set S is convex if, given any two points in S, the points on the line joining them are also in S.

Get comfortable with these concepts if they are unfamiliar.

Unquiz: Prove that the set

$$\{\mathbf{x} \in \mathcal{R}^n : \mathbf{A}\mathbf{x} \geq \mathbf{b}\}$$

is convex, where $\mathbf{A} \in \mathcal{R}^{m \times n}$. []

Some jargon about functions

Sets can be convex, but functions can be, too.

A function f is convex if its graph would hold water: for any points $\mathbf{y}, \mathbf{z}$ in the domain of f, and for all $t \in [0, 1]$,

$$f((1-t)\mathbf{y} + t\mathbf{z}) \leq (1-t)f(\mathbf{y}) + tf(\mathbf{z})\,.$$

In other words, f does not lie above any of its secants.

Picture.

A function f is strictly convex if we can replace $\leq$ by $<$ and $t \in [0, 1]$ by $t \in (0, 1)$ in the definition above.

A function f is concave if $-f$ is convex.

A straight line is both convex and concave!

Example: The function $f(x) = x^2$ is convex, since for $t \in [0, 1]$,

$$\begin{aligned} f(\quad (1-t)\quad & y + tz) - ((1-t)f(y) + tf(z)) \\ &= (1-t)^2y^2 + t^2z^2 + 2(1-t)tyz - (1-t)y^2 - tz^2 \\ &= [(1-t)^2 - (1-t)]y^2 + 2(1-t)tyz + [t^2 - t]z^2 \\ &= [t^2 - t]y^2 + 2(1-t)tyz + t(t-1)z^2 \\ &= t(t-1)[y^2 - 2yz + z^2] \\ &= t(t-1)(y-z)^2 \leq 0\,. \end{aligned}$$

[]

Exercise: Show that $f(x) = |x|$ is convex but not strictly convex.

Example: The function $f(x) = x^3$ is convex for $x > 0$ but concave for $x < 0$. The point $x = 0$ is an inflection point.

Why convexity is important to us

A function is proper convex if it is convex, bounded below, and not identically equal to $+\infty$.

(We may be sloppy and just say "convex" when we mean "proper convex.")

The point $\mathbf{x}^* \in S$ is a local minimizer of f if there exists a number $\epsilon > 0$ such that if $\mathbf{y} \in S$ and $\|\mathbf{y} - \mathbf{x}^*\| < \epsilon$, then $f(\mathbf{x}^*) \leq f(\mathbf{y})$.

The point $\mathbf{x}^* \in S$ is a global minimizer of f on S if, for all points $\mathbf{y} \in S$, $f(\mathbf{x}^*) \leq f(\mathbf{y})$.

Theorem:

- If f is proper convex and if $\mathbf{x}^*$ is a local minimizer of f on a convex set S, then $\mathbf{x}^*$ is the global minimizer on S.
- The set of global minimizers of a proper convex function f on a convex set S is convex.

Proof: For the first part, suppose $\hat{\mathbf{x}}$ is the global minimizer. Then $f(\mathbf{x}^*) > f(\hat{\mathbf{x}})$, so for $t \in [0, 1)$,

$$(1-t)f(\hat{\mathbf{x}}) + tf(\mathbf{x}^*) < f(\mathbf{x}^*)$$

But by convexity,

$$f((1-t)\hat{\mathbf{x}} + t\mathbf{x}^*) \leq (1-t)f(\hat{\mathbf{x}}) + tf(\mathbf{x}^*),$$

so $\mathbf{x}^*$ cannot be a local minimizer. This is a contradiction.

For the second part, it is actually simpler to prove a more general result: the level sets $T_\alpha = \{\mathbf{x} \in S : f(\mathbf{x}) \leq \alpha\}$ are convex. This is true since, for all $\mathbf{y}, \mathbf{z} \in T_\alpha$,

$$f((1-t)\mathbf{y} + t\mathbf{z}) \leq (1-t)f(\mathbf{y}) + tf(\mathbf{z}) \leq \alpha\,,$$

so $(1-t)\mathbf{y} + t\mathbf{z} \in T_\alpha$. []

So if we have a problem involving minimizing a convex function over a convex set, it is much easier to solve than the general problem!

The basic algorithm

The basic algorithm

Our basic strategy is inspired by the foggy mountain:

Take an initial guess at the solution $\mathbf{x}^{(0)}$, our starting point on the mountain. Set $k = 0$.

Until $\mathbf{x}^{(k)}$ is a good enough solution,

Find a search direction $\mathbf{p}^{(k)}$.
Set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}$, where α_k is a scalar chosen to guarantee that progress is made.
Set $k = k + 1$.

Initially, we will study algorithms for which $\alpha_k = 1$.

Unresolved details:

- testing convergence.
- finding a search direction.
- computing the step-length α_k.

The model method: Newton

Newton's method

Newton's method is one way to determine the search direction $\mathbf{p}^{(k)}$. It is inspired by our Taylor series expansion

$$f(\mathbf{x}+\mathbf{p}) \approx f(\mathbf{x}) + \mathbf{p}^T \mathbf{g}(\mathbf{x}) + \frac{1}{2}\mathbf{p}^T \mathbf{H}(\mathbf{x})\mathbf{p} \equiv \hat{f}(\mathbf{p}).$$

Suppose we replace $f(\mathbf{x}+\mathbf{p})$ by the quadratic model $\hat{f}(\mathbf{p})$ and minimize that.

In general, the model won't fit f well at all ... except in a neighborhood of the point $\mathbf{x}$ where it is built. But if our step $\mathbf{p}$ is not too big, that is ok!

So let's try to minimize $\hat{f}$ with respect to $\mathbf{p}$. If we set the derivative equal to zero

$$\mathbf{g}(\mathbf{x}) + \mathbf{H}(\mathbf{x})\mathbf{p} = \mathbf{0}$$

we see that we need the vector $\mathbf{p}$ defined by

$$\mathbf{H}(\mathbf{x})\mathbf{p} = -\mathbf{g}(\mathbf{x})\,.$$

This vector is called the Newton direction, and it is obtained by solving the linear system involving the Hessian matrix and the negative gradient.

Picture.

Note that if the Hessian $\mathbf{H}(\mathbf{x})$ is positive definite, then this linear system is guaranteed to have a unique solution (since $\mathbf{H}(\mathbf{x})$ is nonsingular) and, in addition,

$$0 < \mathbf{p}^T \mathbf{H}(\mathbf{x})\mathbf{p} = -\mathbf{g}(\mathbf{x})^T \mathbf{p},$$

so in this case $\mathbf{p}$ is a downhill direction.

If $\mathbf{H}(\mathbf{x})$ fails to be positive definite, then the situation is not as nice.

- We may fail to have a solution to the linear system.
- We may walk uphill.

We can also get into trouble if $\mathbf{H}(\mathbf{x})$ is close to singular, since in that case it will be difficult to get a good solution to the linear system using floating point arithmetic, so the computed direction may fail to be downhill.

Three easy pictures

We illustrate nice and not-nice quadratic models for $n = 2$ variables.

We'll draw contour plots of the quadratic model $\hat{f}(\mathbf{p})$. (These are like topographical maps. We draw lines defining sets $\{\mathbf{p} : \hat{f}(\mathbf{p}) = c\}$ for some constants c.) These are called level curves.

Let's try to understand what controls the shape of the quadratic model.

Suppose that we solve the eigenvalue problem for $\mathbf{H}(\mathbf{x})$:

$$\mathbf{H}(\mathbf{x})\mathbf{u}_1 = \lambda_1 \mathbf{u}_1 \,, \quad \mathbf{H}(\mathbf{x})\mathbf{u}_2 = \lambda_2 \mathbf{u}_2 \,,$$

where the λ_i are (positive) numbers and the $\mathbf{u}_i$ are vectors that are orthogonal to each other, normalized so that $\mathbf{u}_1^T \mathbf{u}_1 = \mathbf{u}_2^T \mathbf{u}_2 = 1$.

Now $\mathbf{u}_1$ and $\mathbf{u}_2$ form a basis for $\mathcal{R}^2$, so any vector can be expressed as a combination of these two. Let $\mathbf{p} = \alpha_1 \mathbf{u}_1 + \alpha_2 \mathbf{u}_2$ and $\mathbf{g}(\mathbf{x}) = \mu_1 \mathbf{u}_1 + \mu_2 \mathbf{u}_2$.

Our quadratic model becomes

$$\begin{aligned} f(\mathbf{x}+\mathbf{p}) &= f(\mathbf{x}) + \mathbf{p}^T \mathbf{g}(\mathbf{x}) + \frac{1}{2}\mathbf{p}^T \mathbf{H}(\mathbf{x})\mathbf{p} \\ &= f(\mathbf{x}) + \alpha_1\mu_1 + \alpha_2\mu_2 + \frac{1}{2}(\lambda_1\alpha_1^2 + \lambda_2\alpha_2^2)\,. \end{aligned}$$

Setting the derivative (with respect to α_1 and α_2) equal to zero, we obtain

$$\alpha_1 = -\mu_1/\lambda_1 \quad \alpha_2 = -\mu_2/\lambda_2\,.$$

Is this a minimizer of the function? The Hessian is

$$\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix},$$

and this is positive definite iff $\lambda_1, \lambda_2 > 0$.

Case 1: **H**(**x**) is positive definite.

Picture.

In this case, the contours are ellipsoids, the function itself has the shape of a bowl, and the Newton direction **p** points to the bottom of the bowl. The lengths of the axes are proportional to $1/\sqrt{\lambda_1}$ and $1/\sqrt{\lambda_2}$.

Case 2: **H** is positive semi-definite. In this case, one eigenvalue $\lambda_2 = 0$. The contours have become so elongated that they fail to close.

We want to define our solution **p** by

$$\alpha_1 = -\mu_1/\lambda_1 \quad \alpha_2 = -\mu_2/\lambda_2 \, .$$

The definition for α_1 is ok, but we are in trouble for α_2 unless $\mu_2 = 0$; i.e., unless **g** has no component in the direction $\mathbf{u}_2$.

If $\mu_2 = 0$

Picture. Note that the function has the shape of a parabola if we walk orthogonal to a contour.

If **g** does have a component in the $\mathbf{u}_2$ direction, then there is no solution to our quadratic model - there is no finite minimizer.

Picture.

Case 3: **H** is indefinite In this case, $\lambda_1 > 0$ and $\lambda_2 < 0$. The function now has a saddlepoint but no finite minimizer.

Picture.

Bottom line:

To run the basic Newton method successfully, we need the Hessian $\mathbf{H}(\mathbf{x})$ to be positive definite everywhere we need to evaluate it.

Later, we will need to put in safeguards to handle these bad cases when $\mathbf{H}$ fails to be positive definite, but for now, we'll just study the basic Newton algorithm, in which we step from $\mathbf{x}$ to $\mathbf{x} - \mathbf{H}(\mathbf{x})^{-1}\mathbf{g}(\mathbf{x})$.

How well does the Newton Method work?

When it is good, it is very, very good!

Let $\mathbf{e}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^*$ be the error at iteration k.

Theorem: (Fletcher, p46) Suppose $f \in \mathcal{C}^2(S)$ and there is a positive scalar λ such that

$$\|\mathbf{H}(\mathbf{x}) - \mathbf{H}(\mathbf{y})\| \leq \lambda \|\mathbf{x} - \mathbf{y}\|$$

for all points $\mathbf{x}, \mathbf{y}$ in a neighborhood of $\mathbf{x}^*$. Then if $\mathbf{x}^{(k)}$ is sufficiently close to $\mathbf{x}^*$ and if $\mathbf{H}(\mathbf{x}^*)$ is positive definite, then there exists a constant c such that

$$\|\mathbf{e}^{(k+1)}\| \leq c\|\mathbf{e}^{(k)}\|^2 .$$

Proof: Use Taylor series

$$\mathbf{0} = \mathbf{g}(\mathbf{x}^{(\mathbf{k})} - \mathbf{e}^{(\mathbf{k})}) = \mathbf{g}(\mathbf{x}^{(\mathbf{k})}) - \mathbf{H}^{(\mathbf{k})}\mathbf{e}^{(\mathbf{k})} + \mathbf{O}(\|\mathbf{e}^{(\mathbf{k})}\|^{\mathbf{2}}) .$$

We multiply by $(\mathbf{H}^{(k)})^{-1}$. Why is this guaranteed to exist?

$$\mathbf{0} = -\mathbf{p}^{(\mathbf{k})} - \mathbf{e}^{(\mathbf{k})} + \mathbf{O}(\|\mathbf{e}^{(\mathbf{k})}\|)^{\mathbf{2}}) .$$

Now

$$-\mathbf{p}^{(k)} - \mathbf{e}^{(k)} = (\mathbf{x}^{(k)} - \mathbf{x}^{(k+1)}) - (\mathbf{x}^{(k)} - \mathbf{x}^*) = -\mathbf{e}^{(k+1)} ,$$

so $\|\mathbf{e}^{(k+1)}\| = O(\|\mathbf{e}^{(k)}\|^2)$. []

This rate of convergence is called quadratic convergence and it is remarkably fast. If we have an error of 10^{-1} at some iteration, then two iterations later the error will be about 10^{-4} (if $c \approx 1$). After four iterations it will be about 10^{-16}, as many figures as we carry in double precision arithmetic!

How close to Newton do we need to be in order to get fast convergence?

How close to Newton do we need to be?

Definition: A sequence of errors $\mathbf{e}^{(k)}$ converges to zero with rate r and rate constant c if

$$\lim_{k \to \infty} \frac{\|\mathbf{e}^{(k+1)}\|}{\|\mathbf{e}^{(k)}\|^r} = c$$

(If $r = 1$, then c should be < 1.)

Newton's quadratic rate of convergence is nice, but Newton's method is not an ideal method:

- It requires the computation of $\mathbf{H}$ at each iteration.
- It requires the solution of a linear system involving $\mathbf{H}$.
- It can fail if $\mathbf{H}$ fails to be positive definite.

So we would like to modify Newton's method to make it cheaper and more widely applicable without sacrificing its fast convergence.

An important result

We can get superlinear convergence (convergence with rate $r > 1$) without walking exactly in the Newton direction:

Theorem: (N&S p304) Suppose

- f is defined on an open convex set S with minimizer $\mathbf{x}^* \in S$.
- There exists a finite constant ℓ such that

  $$\|\mathbf{H}(\mathbf{x}) - \mathbf{H}(\mathbf{y})\| \leq \ell \|\mathbf{x} - \mathbf{y}\|$$

  for all $\mathbf{x}, \mathbf{y} \in S$.
- We compute a sequence $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{p}^{(k)}$ so that each iterate is in S, and none of them equals $\mathbf{x}^*$.
- $\mathbf{H}(\mathbf{x}^*)$ is positive definite.

Then the sequence $\{\mathbf{x}^{(k)}\} \to \mathbf{x}^*$ superlinearly if and only if

$$\lim_{k \to \infty} \frac{\|\mathbf{p}^{(k)} + \mathbf{H}(\mathbf{x}^{(k)})^{-1} \mathbf{g}(\mathbf{x}^{(k)})\|}{\|\mathbf{p}^{(k)}\|} = 0 \,.$$

Proof: See N&S. []

This enables us to

- fix the Newton method, when the Hessian fails to behave, without destroying the convergence rate.
- incorporate some shortcuts to make each iteration cheaper.

We'll postpone the discussion of shortcuts (quasi-Newton methods) until later.

Making the Newton method safe

When does Newton get into trouble?

We want to modify Newton whenever we are not sure that the direction it generates is downhill.

If the Hessian is positive definite, we know the direction will be downhill, although if $\mathbf{H}$ is nearly singular, we may have some computational difficulties.

If the Hessian is semidefinite or indefinite, we might or might not get a downhill direction.

Our strategy:

- We'll use the Hessian matrix whenever it is positive definite and not close to singular, because it leads to quadratic convergence.
- We'll replace $\mathbf{H}(\mathbf{x})$ by $\hat{\mathbf{H}}(\mathbf{x}) = \mathbf{H}(\mathbf{x}) + \hat{\mathbf{E}}$ whenever $\mathbf{H}$ is close to singularity or fails to be positive definite.

Conditions on $\hat{\mathbf{H}}$:

- $\hat{\mathbf{H}}$ is symmetric positive definite.
- $\hat{\mathbf{H}}$ is not too close to singular; in other words, its smallest eigenvalue is bounded below by a constant bigger than zero.

Modifying Newton's method

Reference: GMW Chap 4, Murray Chap 4, GM Chap 2.

Strategy 1: Greenstadt's method. If some eigenvalue is negative or too close to zero, replace it by δ, where

$$\delta = \max(2^{-t}\|\mathbf{H}\|_\infty, 2^{-t}),$$

where 2^{-t} is machine epsilon, the gap between 1 and the next larger floating-point number, and

$$\|\mathbf{H}\|_\infty = \max_{i=1,\ldots,n} \sum_{j=1}^{n} |h_{ij}| \,.$$

This gives a matrix $\hat{\mathbf{H}}$ that is positive definite and has bounded condition number.

Greenstadt's method was the first one. It is very effective but a lot of work!

Strategy 2: Levenberg-Marquardt method. This one was actually proposed for least squares problems, but it works here, too.

Replace $\mathbf{H}$ by

$$\hat{\mathbf{H}} = \mathbf{H} + \gamma \mathbf{I}\,.$$

This shifts every eigenvalue up by γ.

How do we choose γ? It is usually done by trial and error: seek a γ so that $\hat{\mathbf{H}}$ is positive definite and $\|\mathbf{p}^{(k)}\| \leq h^{(k)}$ where $\{h^{(k)}\}$ is a given sequence of numbers.

Note: If the h's are small enough, then we can avoid using a line search. (Line searches will be discussed later, but their disadvantage is that they require the function to be evaluated many times.)

Strategy 3: Bad Strategies: Note that there are many numerically unstable alternatives in the literature. Beware!

Strategy 4: Cholesky Strategies: developed by Gill and Murray.

Background: Any symmetric positive definite matrix $\mathbf{A}$ can be factored as

$$\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{L}^T$$

where $\mathbf{D}$ is a diagonal matrix and $\mathbf{L}$ is lower triangular with ones on its main diagonal.

Example:

$$\begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{21} & a_{22} & a_{32} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \ell_{21} & 1 & 0 \\ \ell_{31} & \ell_{32} & 1 \end{bmatrix} \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix} \begin{bmatrix} 1 & \ell_{21} & \ell_{31} \\ 0 & 1 & \ell_{32} \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} d_1 & d_1\ell_{21} & d_1\ell_{31} \\ d_1\ell_{21} & d_1\ell_{21}^2 + d_2 & d_1\ell_{21}\ell_{31} + d_2\ell_{32} \\ d_1\ell_{31} & d_1\ell_{21}\ell_{31} + d_2\ell_{32} & d_1\ell_{31}^2 + d_2\ell_{32}^2 + d_3 \end{bmatrix}$$

so, from the 1st column we obtain

$$\begin{aligned} d_1 &= a_{11} \\ \ell_{21} &= a_{21}/d_1 \\ \ell_{31} &= a_{31}/d_1 \end{aligned}$$

and from the second

$$\begin{aligned} d_2 &= a_{22} - d_1\ell_{21}^2 \\ \ell_{32} &= (a_{32} - d_1\ell_{21}\ell_{31})/d_2\,. \end{aligned}$$

From the last column, we obtain

$$d_3 = a_{33} - d_1\ell_{31}^2 - d_2\ell_{32}^2\,,$$

completing the factorization. []

Properties of the Cholesky factorization:

- Set $\mathbf{U} = \mathbf{D}\mathbf{L}^T$. Then $\mathbf{U}$ is the matrix obtained by Gauss elimination, without pivoting, on $\mathbf{A}$.
- The factorization is stable if $\mathbf{A}$ is positive definite. This means that there are bounds on the absolute values of elements in $\mathbf{L}$ and $\mathbf{D}$ in terms of the matrix $\mathbf{A}$. (Without this, small errors in $\mathbf{A}$ can cause large errors in the factors.) In particular, the diagonal elements of $\mathbf{D}$ are bounded below.

  Example:

  $$\begin{bmatrix} 1 & a \\ a & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ a & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1-a^2 \end{bmatrix} \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}$$

  []
- If $\mathbf{A}$ is not positive definite, then we get a zero or negative element on the diagonal of $\mathbf{D}$.
- The cost of Cholesky is $n^3/6 + O(n^2)$ multiplications, about half the cost of Gauss elimination.

Modified Cholesky algorithms

Idea:

- While factoring, if any $d_{ii} \leq 0$, modify it so that it is positive. This changes the factored matrix from $\mathbf{H}$ to $\hat{\mathbf{H}}$.
- If modification is needed, try to keep $\|\mathbf{H} - \hat{\mathbf{H}}\|$ small so that we will have an almost-Newton direction.

- To keep close to Newton, we want $\|\mathbf{H} - \hat{\mathbf{H}}\| = 0$ if $\mathbf{H}$ is positive definite, and we want $\hat{\mathbf{H}}$ to be a continuous function of $\mathbf{H}$.
- Making $\|\mathbf{H} - \hat{\mathbf{H}}\| = 0$ if $\mathbf{H}$ is positive definite is not really possible, since we also need to modify $\mathbf{H}$ if any eigenvalue is positive but too close to zero.
- We choose to make $\hat{\mathbf{H}} = \mathbf{H} + \mathbf{E}$, where $\mathbf{E}$ is diagonal.

Three ways to modify H using Cholesky

1. $\hat{\mathbf{E}} = \gamma \mathbf{I}$ for some $\gamma \geq 0$. This is akin to Levenberg-Marquardt.
2. $\hat{\mathbf{E}}$ = a general diagonal matrix computed in the course of the Cholesky factorization.

   Reference: For more details, see GMW pp. 109-111, or FOL.

3. Forsgren, Gill, Murray (FGM)

   Perform Cholesky factorization with diagonal pivoting, permuting the matrix at each step to put the largest remaining diagonal element to the pivot position.

   This postpones the modification and keeps it as small as possible.

   Accept the main diagonal element if it is $\geq \nu$ times the largest absolute value of others in its row. (ν is a parameter between 0 and 1).

   When no acceptable element remains, we have

$$\begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{D}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2 \end{bmatrix} \begin{bmatrix} \mathbf{L}_{11}^T & \mathbf{L}_{21}^T \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$

   where $\mathbf{D}_1$ is diagonal but $\mathbf{D}_2$ is not.

   Now replace $\mathbf{D}_2$ by a positive definite matrix $\hat{\mathbf{D}}_2$ and complete the factorization.

   For this algorithm, there are nice bounds: If $\hat{\mathbf{H}}\mathbf{p} = -\mathbf{g}$, then

   - The inner product of $\mathbf{p}$ with $-\mathbf{g}$ is bounded below, so we can't approach orthogonality with $\mathbf{g}$.
   - The size of $\mathbf{p}$ is bounded above, so we don't get unreasonably large directions.

   Note: This algorithm takes no extra work if we arrange the Cholesky factorization in outer product form.

Unquiz: Write out the FGM algorithm. []

A bonus from these modification methods

In addition to providing a descent direction in case $\mathbf{H}$ is indefinite, the Cholesky methods also provide one if we are at a stationary point ($\mathbf{g} = \mathbf{0}$) that is a saddle point instead of a minimizer.

In this case, we cannot use the Newton-like direction, since it is zero.

What do we do?

Taylor series says

$$f(\mathbf{x}+\mathbf{p}) = f(\mathbf{x}) + \mathbf{g}^T\mathbf{p} + \frac{1}{2}\mathbf{p}^T\mathbf{H}(\mathbf{x})\mathbf{p} + O(\|\mathbf{p}\|^3)\,.$$

Greenstadt method: Choose

$$\mathbf{p} = \sum_{i:\lambda_i<0} \alpha_i \mathbf{u}_i$$

where the α_i are any scalars. Then

$$\mathbf{p}^T\mathbf{H}(\mathbf{x})\mathbf{p} = \sum_{i:\lambda_i<0} \alpha_i^2\lambda_i < 0\,.$$

Cholesky methods:

A direction of negative curvature can be computed from the modified part of the factorization. For the Forsgren et al. method, for example, a direction can be determined by solving

$$\mathbf{L}^T\mathbf{p} = \mathbf{r}$$

where $\mathbf{r}$ is a vector with at most two nonzero entries: $\mathbf{r} = \mathbf{e}_q$, if d_{qq} has the largest absolute value of any entry in $\mathbf{D}_2$, and $\mathbf{r} = \mathbf{e}_q - sign(d_{qs})\mathbf{e}_s$ if d_{qs} is the largest. Then

$$\mathbf{p}^T\mathbf{H}\mathbf{p} = \mathbf{p}^T\mathbf{L}\mathbf{D}\mathbf{L}^T\mathbf{p} - \mathbf{p}^T\mathbf{E}\mathbf{p} < 0$$

since both terms are negative.

Therefore, any of these methods can be used to obtain a direction of negative curvature in case we arrive at a saddle point.

What our algorithm now looks like

Recall: Until $\mathbf{x}^{(k)}$ is a good enough solution,

Find a search direction $\mathbf{p}^{(k)}$.
Set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}$, where α_k is a scalar chosen to guarantee that progress is made.

Now we have some details for Newton's method.
Find a search direction $\mathbf{p}^{(k)}$ means

> Calculate $\mathbf{g}^{(k)}$, $\mathbf{H}^{(k)}$.
> Factor $\mathbf{H}^{(k)} = \mathbf{L}\hat{\mathbf{D}}\mathbf{L}^T - \hat{\mathbf{E}}$.
> If $\|\mathbf{g}^{(k)}\| < \epsilon$ and $\hat{\mathbf{E}} = \mathbf{0}$, then halt with an approximate solution.
> Otherwise find a direction:
>
> > If $\|\mathbf{g}^{(k)}\| > \epsilon$, then solve $\mathbf{L}\hat{\mathbf{D}}\mathbf{L}^T\mathbf{p}^{(k)} = -\mathbf{g}^{(k)}$ to get a downhill direction.
> > Otherwise get a direction of negative curvature by solving $\mathbf{L}^T\mathbf{p}_k = \mathbf{r}$, with $\mathbf{r}$ defined earlier.

What is missing? How long a step should we take in the direction $\mathbf{p}$?

Descent directions and line searches.

A backtracking line search

Reference: Dennis & Schnabel pp. 6-17. Algorithm due to Dennis, More', Schnabel.

We take

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{p}^{(k)} .$$

How do we choose $\alpha^{(k)}$?

Let

$$F(\alpha) = f(\mathbf{x} + \alpha\mathbf{p}) .$$

Then

$$F'(\alpha) = \mathbf{p}^T\mathbf{g}(\mathbf{x} + \alpha\mathbf{p}) .$$

Backtracking line search:

> Choose $\alpha = 1$ (to give the full Newton step).
> While α is not good enough,
>
> > Choose a new $\alpha_{new} \in [0, \alpha]$ by interpolation, and set $\alpha = \alpha_{new}$.

Note: If **p** is not the Newton direction, then we may need an initial braketing phase to find a good upper bound on α by testing larger values.

Reference: See Fletcher Section 2.6 for details.

How do we do the interpolation?

- Initially, we know three pieces of information: $F(0) = f(\mathbf{x}^{(k)})$, $F'(0) = \mathbf{p}^{(k)T}\mathbf{g}(\mathbf{x}^{(k)})$, and $F(1)$.

  Three pieces of data determine a quadratic model for F:

  $$F_q(\lambda) = [F(1) - F(0) - F'(0)]\lambda^2 + F'(0)\lambda + F(0)\,.$$

  The minimizer is

  $$\alpha_1 = -\frac{F'(0)}{2[F(1) - F(0) - F'(0)]}\,,$$

  so we take

  $$\alpha_{new} = \max(\alpha_1, 0.1)$$

  (or substitute some other tolerance for 0.1).

- Later, we have four recent pieces of information: $F(0)$, $F'(0)$, $F(\alpha)$, $F(\alpha_{old})$. We already know that a quadratic model does not fit well, so we try a cubic:

  $$F_c(\lambda) = a\lambda^3 + b\lambda^2 + F'(0)\lambda + F(0)\,.$$

  The minimizers are

  $$\alpha_\pm = \frac{-b \pm \sqrt{b^2 - 3aF'(0)}}{3a}$$

  and we take

  $$\alpha_{new} = \max(\min(\alpha_+, \alpha_-), 0.1)$$

  if $\alpha_\pm$ is real, and

  $$\alpha_{new} = re(\alpha_+) = \max(-b/3a, 0.1)$$

  otherwise.

How do we decide that α is good enough?

Reference: N&S 10.5, Fletcher pp.26ff.

Example: why descent is not enough. Let $f(x) = x^2$, $x^{(0)} = 2$,

$$x^{(k)} = \frac{2k+1}{2k}$$

Then the sequence $\{x^{(k)}\} = 2, 3/2, 5/4, \ldots$ gives decreasing values for f but converges to 1. []

Therefore, we need stronger conditions than descent in order to guarantee convergence.

Our situation

- Have a downhill direction **p**, so we know that for very small α, $F(\alpha) < F(0)$.
- If **p** = the Newton direction, then we predict that $\alpha = 1$ is the minimizer.
- We want an upper bound on the αs to consider, since Newton's method is based on a quadratic model and is not expected to fit the function well if we go too far.
- We might have F' available.
- We really can't afford an exact line search. In an exact linesearch we find the value of α that exactly minimizes $f(\mathbf{x} + \alpha\mathbf{p})$. We can do this for quadratic functions, since in that case a formula for α can be derived, but in general exact linesearch is impossible and is only interesting because a lot of theorems demand it.

What do we do?

First idea: Goldstein conditions

Goldstein (1965)

A step $\alpha\mathbf{p}$ is acceptable if

1. It gives sufficent decrease relative to its size: $F(\alpha) < F(0) + \alpha\rho F'(0)$ for some fixed $\rho \in (0, 1/2)$.
2. It is not too small: $F(\alpha) > F(0) + \alpha(1-\rho)F'(0)$. (For small α, $F(\alpha) \approx F(0) + \alpha F'(0)$, so these points are unacceptable.)
3. The direction is downhill and bounded away from orthogonality to **g**: $\mathbf{g}^T\mathbf{p} \leq -\delta\|\mathbf{g}\|\|\mathbf{p}\|$ for some fixed $\delta > 0$.

Picture.

Second idea: Wolfe(1968)-Powell(1976) conditions

A step $\alpha\mathbf{p}$ is acceptable if

1. It gives sufficent decrease relative to its size: $F(\alpha) < F(0) + \alpha\rho F'(0)$ for some fixed $\rho \in (0, 1/2)$.
2. NEW! It is not too small: $F'(\alpha) \geq \sigma F'(0)$ for some fixed $\sigma \in (\rho, 1)$. Disadvantage over Goldstein: requires derivative evaluation at each step.
3. The direction is downhill and bounded away from orthogonality to $\mathbf{g}$: $\mathbf{g}^T\mathbf{p} \leq -\delta\|\mathbf{g}\|\|\mathbf{p}\|$ for some fixed $\delta > 0$.

Picture.

What do these conditions buy for us?

It can be shown that acceptable points exist as long as the minimizer is finite.

Typical theorem: Global convergence of descent methods. If

- f is continuously differentiable and bounded below,
- $\mathbf{g}$ is Lipschitz continuous for all $\mathbf{x}$, i.e., there exists a constant L such that, for all $\mathbf{x}$ and $\mathbf{y}$,
$$\|\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$$

Then either $\mathbf{g}^{(k)} = \mathbf{0}$ for some k or $\mathbf{g}^{(k)} \to \mathbf{0}$.

Proof: See Fletcher.

Reference: There is a similar (but less satisfying) theorem on p. 316 of N&S.

Unquiz: Write the algorithm for a backtracking linesearch satisfying the Wolfe conditions.

[]

Trust regions.

Trust regions: an alternative to linesearch

Reference: Fletcher Chapter 5.

Trust region methods determine α and $\mathbf{p}$ simultaneously.

Idea: Use $\mathbf{g}$ and $\mathbf{H}$ to form a quadratic model

$$f(\mathbf{x} + \mathbf{p}) \approx q(\mathbf{p}) = f(\mathbf{x}) + \mathbf{p}^T\mathbf{g} + \frac{1}{2}\mathbf{p}^T\mathbf{H}\mathbf{p}\,.$$

But we should only trust the model when $\|\mathbf{p}\| < h$ for some small scalar h.

Let $\mathbf{x}_{new} = \mathbf{x} + \mathbf{p}^*$ where $\mathbf{p}^*$ solves

$$\min_{\|\mathbf{p}\| \leq h} q(\mathbf{p}) \,.$$

Note: Depending on the norm we choose, this gives us different geometries for the feasible set defined by $\|\mathbf{p}\| < h$.

Still to be determined:

- How to determine h and adapt it.
- How to find $\mathbf{p}^*$.

How to find p^*

The answer changes, depending on norm we choose.

Suppose we choose the infinity norm:

$$\min_{|p_i| \leq h} q(\mathbf{p}) \,.$$

This is a quadratic programming problem with bound constraints. We'll study algorithms for it later.

Unquiz: Solve the trust region problem for the 2-norm:

$$\min_{\mathbf{p}^T\mathbf{p} \leq h^2} q(\mathbf{p}) \,,$$

$$q(\mathbf{p}) = f(\mathbf{x}) + \mathbf{p}^T\mathbf{g} + \frac{1}{2}\mathbf{p}^T\mathbf{H}\mathbf{p} \,,$$

using Lagrange Multipliers.

Note the relationship to the Levenberg-Marquardt algorithm. []

Picture: Dogleg step.

How to choose h

Idea: h determines the region in which our model q is known to be a good approximation to f:

$$r \equiv \frac{f(\mathbf{x}+\mathbf{p}) - f(\mathbf{x})}{q(\mathbf{p}) - q(\mathbf{0})} \approx 1 \,.$$

Heuristic suggested by Powell:

- If r too small ($< 1/4$) then reduce h by a factor of 4.
- If r close to 1 ($> 3/4$) then increase h by a factor of 2.

Note that this can be done by modifying γ, the parameter in the Levenberg-Marquardt algorithm.

Pitfall in trust region methods: If the problem is poorly scaled, then the trust region will remain very small and we will never be able to take large steps to get us close to the solution.

Example: $f(\mathbf{x}) = f_1(x_1) + f_1(10000x_2)$ where f_1 is a well-behaved function. []

Convergence of trust region methods

Typical theorem: Global convergence of trust region methods.
If

- $S = \{\mathbf{x} : f(\mathbf{x}) \leq f(\mathbf{x}^{(0)})\}$ is bounded.
- $f \in \mathcal{C}^2(S)$

Then the sequence $\{\mathbf{x}^{(k)}\}$ has an accumulation point $\mathbf{x}^*$ that satisfies the first- and second-order necessary conditions for optimality.

Final words

We now know how to recognize a solution and compute a solution using Newton's method.

We have added safeguards in case the Hessian fails to be positive definite, and we have added a linesearch to guarantee convergence.

The resulting algorithm converges rather rapidly, but each iteration is quite expensive.

Next, we want to investigate algorithms that have lower cost per iteration.

References

- Dennis and Schnabel: J. E. Dennis Jr and R. B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice Hall 1983
- FGM: Forsgren, Gill, Murray, SIAM J. on Scientific Computing 16 (1995) p139

- Fletcher: Practical Methods of Optimization by Roger Fletcher, Wiley, 1987.
- FOL: Haw-ren Fang and Dianne P. O'Leary, "Modified Cholesky Algorithms: A Catalog with New Approaches," *Mathematical Programming A*, 2007. DOI:10.1007/s10107-007-0177-6
- GMW: Philip E. Gill, Walter Murray, Margaret H. Wright, Practical Optimization, Academic Press, 1981
- GM: P. Gill and W. Murray, eds., Numerical Methods for Constrained Optimization, Academic Press 1974
- Murray: W. Murray, ed., Numerical Methods for Unconstrained Optimization, Academic Press 1972
- N&S: Linear and Nonlinear Programming by Stephen G. Nash and Ariela Sofer, McGraw-Hill 1996.