The Descent Method and Line Searches

Lecture 2, Continuous Optimisation

Oxford University Computing Laboratory, HT 2006
Notes by Dr Raphael Hauser (hauser@comlab.ox.ac.uk)

Chapter I: Unconstrained Optimisation

Unconstrained optimisation deals with problems of the form

$$\text{(P)} \quad \min_{x \in \mathbb{R}^n} f(x)$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuous.

Furthermore, we usually assume that f is C^2 with Lipschitz-continuous Hessian, that is, $\exists\, \Lambda > 0$ such that

$$\|D^2 f(x) - D^2 f(y)\| \leq \Lambda \|x - y\| \quad \forall\, x, y \in \mathbb{R}^n.$$

Example 1: Risk minimisation under shortselling

- Let us go back to Example 2 of Lecture 1. By eliminating $x_n = 1 - \sum_{i=1}^{n-1} x_i$ we can get rid of the constraint

$$\sum_{i=1}^{n} x_i = 1.$$

- Furthermore, if we allow short-selling of assets, the constraints

$$x_i \geq 0 \quad (i = 1, \ldots, n)$$

are no longer imposed.

- Finally, let us suppose all the assets considered have the same expected return $\mu_i \equiv \mu$, so that the only sensible choice for the target return b is μ itself and the constraint

$$\sum_{i=1}^{n} \mu_i x_i \geq b$$

can be omitted.

The investor's aim is to minimise the risk, which can be modelled as

$$\begin{aligned}\min_{x \in \mathbb{R}^{n-1}} \; & f(x_1, \dots, x_{n-1}) \\ &= \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} \sigma_{ij} x_i x_j + \sum_{j=1}^{n-1} \sigma_{nj} \Big(1 - \sum_{i=1}^{n-1} x_i\Big) x_j \\ &+ \sum_{i=1}^{n-1} \sigma_{in} x_i \Big(1 - \sum_{j=1}^{n-1} x_j\Big) + \sigma_{nn} \Big(1 - \sum_{i=1}^{n-1} x_i\Big)\Big(1 - \sum_{j=1}^{n-1} x_j\Big).\end{aligned}$$

- Since the objective function f is a quadratic (degree 2) polynomial in the decision variables $x_1, \ldots, x_{n-1}$, we have $f \in C^\infty$.

- Moreover, the Hessian $D^2 f(x)$ is the same $(n-1) \times (n-1)$ matrix

$$\begin{pmatrix} 1 & & 0 & -1 \\ & \ddots & & -1 \\ 0 & & 1 & -1 \end{pmatrix} \begin{pmatrix} \sigma_{11} & \ldots & \sigma_{1n} \\ & \ddots & \\ \sigma_{n1} & \ldots & \sigma_{nn} \end{pmatrix} \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \\ -1 & \ldots & -1 \end{pmatrix}$$

for all x, and hence $x \mapsto D^2 f(x)$ is a constant function, which is of course Lipschitz-continuous:

$$\|D^2 f(x) - D^2 f(y)\| = 0 \leq 0 \times \|x - y\| \quad \forall\, x, y \in \mathbb{R}^{n-1}.$$

Example 2:

- On a CAD system it takes n parameters $x_1, \ldots, x_n$ to define the shape of a car.

- An engineer has a piece of software which takes the design parameters $x \in \mathbb{R}^n$ as input and computes the air resistance $f(x)$ of the corresponding fuselage as output.

- The software contains typically millions of lines of code, but for theoretical reasons it is known that $f \in C^2$.

- Using *automatic differentiation*, the engineer can automatically produce a piece of software that computes directional derivatives

$$D_v f(x) = \frac{d}{dt} f(x + tv), \qquad D_{u,v} f(x) = \frac{d^2}{ds\, dt} f(x + su + tv).$$

- How to choose the design parameters so as to minimise the drag on the fuselage?

Some Notation:

- If $x \in \mathbb{R}^n$ then $\|x\|$ denotes the Euclidean norm $\sqrt{\sum x_i^2}$.

- If $A : \mathbb{R}^n \to \mathbb{R}^m$ is a linear map, then $\|A\|$ denotes the operator norm defined by the Euclidean norms on $\mathbb{R}^n$ and $\mathbb{R}^m$, that is,

$$\|A\| = \inf\{\lambda > 0 : \|Ax\| \leq \lambda\|x\| \;\forall\, x \in \mathbb{R}^n\}.$$

- The gradient $\nabla f(x)$ of a function $f : \mathbb{R}^n \to \mathbb{R}$ is sometimes denoted by $g_f(x)$, and its Hessian $D^2 f(x)$ by $H_f(x)$.

- The Jacobian $Df(x)$ of a function $f : \mathbb{R}^n \to \mathbb{R}^m$ is sometimes denoted by $J_f(x)$. Note: if $m = 1$ then $J_f(x) = g_f(x)^{\mathsf{T}}$.

Theorem 1: Optimality Conditions for Unconst. Opt.

(i) Necessary first order optimality condition: if $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable at $x^* \in \mathbb{R}^n$ and has a local minimum there, then $\nabla f(x^*) = 0$ (x^* is a *stationary point* of f).

(ii) Necessary second order condition: if $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable at $x^* \in \mathbb{R}^n$ and has a local minimum there, then $D^2 f(x^*)$ is positive semidefinite (i.e., $h^\mathsf{T} D^2 f(x^*) h \geq 0$ for all $h \in \mathbb{R}^n$; we write $D^2 f(x^*) \succeq 0$ to express this).

(iii) Sufficient optimiality conditions: if $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable at $x^* \in \mathbb{R}^n$, and if $\nabla f(x^*) = 0$ and $D^2 f(x^*)$ is positive definite (i.e., $h^\mathsf{T} \nabla^2 f(x^*) h > 0$ for all $h \in \mathbb{R}^n \setminus \{0\}$; we write $D^2 f(x^*) \succ 0$), then x^* is a local minimiser of f.

Simple idea of proof: use Taylor approximation!

- x^* is a local minimiser $\Rightarrow$ there exists $\epsilon > 0$ such that

$$f(x^* + h) \geq f(x^*), \quad \forall h \in B_\epsilon(0),$$

- Therefore, writing $\langle \cdot, \cdot \rangle$ for the Euclidean inner product, $\forall\, h \in \mathbb{R}^n$,

$$\langle \nabla f(x^*), h \rangle = \lim_{t \to 0} \frac{f(x^* + th) - f(x^*)}{t} \geq \lim_{t \to 0} \frac{f(x^*) - f(x^*)}{t} = 0.$$

- In particular, apply this inequality to $h = -\nabla f(x^*)$:

$$0 \leq \langle \nabla f(x^*), -\nabla f(x^*) \rangle = -\|\nabla f(x^*\|^2 \leq 0,$$

- This shows that $\nabla f(x^*) = 0$ and establishes i).

- For proofs of ii) and iii), see the Lecture Note 2. These are based on 2nd order Taylor approximations.

Important Consequence: Solving the simultaneous system of nonlinear equations

$$\nabla f(x) = 0$$

by an iterative procedure generating a sequence of points $(x_k)_{\mathbb{N}}$, if we can assure that $f(x_k)$ decreases in each iteration,

$$f(x_{k+1}) \leq f(x_k) \quad \forall\, k,$$

then in practice $(x_k)_{\mathbb{N}}$ can only converge to a *local minimiser* x^* and

$$\|\nabla f(x^*)\| < \epsilon$$

can be used as a stopping criterion.

There are two main families of such procedures:

1. Line-search methods

2. Trust-region methods

Example 3: Steepest descent without line searches

Starting from some $x_0 \in \mathbb{R}^n$, compute a sequence of intermediate solutions $(x_k)_{\mathbb{N}}$ defined by

$$x_{k+1} = x_k - \nabla f(x_k).$$

- The method is motivated by the fact that $-\nabla f(x_k)$ is the direction in which f decreases fastest when moving away from x_k.

- For small $t > 0$ decrease occurs: $f(x_k - t\nabla f(x_k)) \leq f(x_k)$.

- However, it is not necessarily the case that $f(x_{k+1}) \leq f(x_k)$, as the step $-\nabla f(x_k)$ can be too far.

- To make the method work, *line-searches* are necessary: in each iteration find $t_k > 0$ such that

$$f(x_k - t\nabla f(x_k)) < f(x_k)$$

  and set

$$x_{k+1} = x_k - t\nabla f(x_k).$$

- Warning: although this method works in principle, it is too primitive to produce any good results in practice!

We now set out to generalise this example.

Algorithm 1: Descent method. Choose a starting point $x_0 \in \mathbb{R}^n$ and a tolerance parameter $\epsilon > 0$. Set $k = 0$.

S1 If $\|\nabla f(x_k)\| \le \epsilon$ then stop and output x_k as an approximate minimiser.

S2 Choose a *search direction* $d_k \in \mathbb{R}^n$ such that $\langle \nabla f(x_k), d_k \rangle < 0$.

S3 Choose a step size $\alpha_k > 0$ such that $f(x_k + \alpha_k d_k) < f(x_k)$.

S4 Set $x_{k+1} := x_k + \alpha_k d_k$, replace k by $k+1$, and go to S1.

The generality of Algorithm 1 leaves flexibility both in

1. the choice of the step length α_k,

2. and in the search direction d_k.

In the remainder of this lecture we discuss the step length selection and treat the choice of good search directions in the next few lectures.

Line-Searches:

In an *exact line-search* α_k is defined by

$$\alpha_k := \inf\{\alpha \geq 0 : \phi'(\alpha) = 0\},$$

where $\phi(\alpha) = f(x_k + \alpha d_k)$.

Note that the point $x_k + \alpha_k d_k$ is the first stationary point of f encountered along the half line $\{x_k + \alpha d_k : \alpha \geq 0\}$.

Note that if $\{\alpha \geq 0 : \phi'(\alpha) = 0\} = \emptyset$, as is the case for example when $\phi(\alpha) = -\ln \alpha$, then $\{\alpha \geq 0 : \phi'(\alpha) = 0\} = \emptyset$, and hence $\alpha_k := \inf \emptyset = +\infty$ corresponds to an infinitely long step which is still sensible.

- Exact line searches are mainly a theoretical tool in the convergence analysis of algorithms.

- In practice, they are computationally too expensive.

Let us now derive step length computations that are equally good choices for the purposes of Algorithm 1.

Definition 1: Wolfe Conditions

We say that the step size α_k of Algorithm 1 satisfies the *Wolfe conditions* if

$$\begin{aligned} \phi(\alpha_k) &\leq \phi(0) + c_1 \alpha_k \phi'(0), \quad \text{and} && (1) \\ \phi'(\alpha_k) &\geq c_2 \phi'(0), && (2) \end{aligned}$$

where $0 < c_1 < 1/2$ and $c_1 < c_2 < 1$ are constants, and where ϕ is the function $\phi(\alpha) = f(x_k + \alpha d_k)$.

- Condition (1) ensures that the actual objective value decrease $f(x_k) - f(x_k + \alpha_k d_k)$ equals at least a fixed fraction of the change $-\alpha_k \langle \nabla f(x_k), d_k \rangle$ predicted by the first order Taylor approximation

$$f(x_k + \alpha_k d_k) \approx f(x_k) + \alpha_k \langle \nabla f(x_k), d_k \rangle.$$

- The restriction $c_1 \leq 1/2$ is desirable because this allows α_k to take the value of the exact minimiser when $\phi(\alpha)$ is a convex quadratic function.

- Condition (2) on the other hand guarantees that the step size is not zero, because $\langle \nabla f(x_k + \alpha_k d_k), d_k \rangle$ is substantially larger than $\langle \nabla f(x_k), d_k \rangle$ (which is a negative number).

Proposition 1: Feasible Step Length Exists

If $f \in C^1(\mathbb{R}^n)$ is bounded below on the half-line $\{x_k + \alpha d_k : \alpha \geq 0\}$ then there exists a step length $\alpha_k \in (0, \infty)$ that satisfies the Wolfe conditions.

Proof: See Lecture Note 2.

Convergence Analysis of Descent Methods

Lemma 1

Let Algorithm 1 be applied to a C^1 function f with Λ-Lipschitz continuous gradient and assume that the step length α_k satisfies the Wolfe conditions. Then

$$f(x_{k+1}) \leq f(x_k) - c_1(1-c_2)\frac{(\cos^2\theta_k)\|\nabla f(x_k)\|^2}{\Lambda},$$

where θ_k is the angle between d_k and $-\nabla f(x_k)$, and where c_1, c_2 are the constants from Definition 1.

- The second Wolfe condition implies

$$\begin{aligned}\langle \nabla f(x_k + \alpha_k d_k), d_k \rangle - \langle \nabla f(x_k), d_k \rangle &= \phi'(\alpha_k) - \phi'(0) \\ &\geq (c_2 - 1)\phi'(0) \\ &= (1 - c_2)\left(-\langle \nabla f(x_k), d_k \rangle\right).\end{aligned}$$

- The Cauchy–Schwartz inequality and the Lipschitz condition imply that the left hand side of this expression is bounded above by $\alpha_k \Lambda \|d_k\|^2$.

- Therefore,

$$\alpha_k \geq (1 - c_2) \cdot \frac{-\langle \nabla f(x_k), d_k \rangle}{\Lambda \|d_k\|^2}.$$

- Since $\langle \nabla f(x_k), d_k \rangle < 0$, Condition (1) yields

$$f(x_{k+1}) = \phi(\alpha_k) \leq \phi(0) + c_1 \alpha_k \phi'(0)$$
$$\leq f(x_k) - c_1(1 - c_2) \frac{\Big(\langle \nabla f(x_k), d_k \rangle\Big)^2}{\Lambda \|d_k\|^2}.$$

- Since

$$\langle \nabla f(x_k), d_k \rangle = -\cos \theta_k \|d_k\| \cdot \|\nabla f(x_k)\|,$$

this proves the result.

Theorem 2: Convergence of Descent Method

Suppose $f \in C^1(\mathbb{R}^n)$ has Lipschitz continuous gradients on $\mathbb{R}^n$ and is bounded below. When Algorithm 1 is applied with step lengths α_k that satisfy the Wolfe conditions then

$$\sum_{k=0}^{\infty} (\cos^2 \theta_k) \|\nabla f(x_k)\|^2 < \infty,$$

where θ_k is defined as in Lemma 1.

- Let b be a lower bound for f, that is $f(x) \geq b$ for all $x \in \mathbb{R}^n$.

- Lemma 1 shows that

$$\begin{aligned} f(x_0) - b &\geq f(x_0) - f(x_{k+1}) \\ &\geq f(x_0) - f(x_k) + c_1(1 - c_2)\frac{(\cos^2\theta_k)\|\nabla f(x_k)\|^2}{\Lambda} \\ &\geq \dots \\ &\geq f(x_0) - f(x_0) + \frac{c_1(1 - c_2)}{\Lambda}\sum_{k=0}^{j}(\cos^2\theta_k)\|\nabla f(x_k)\|^2. \end{aligned}$$

- Therefore,

$$0 \leq \sum_{k=0}^{j}(\cos^2\theta_k)\|\nabla f(x_k)\|^2 \leq \frac{\big(f(x_0) - b\big)\Lambda}{c_1(1 - c_2)}.$$

Theorem 2 establishes that

- either $\nabla f(x_k)$ converges to the zero vector as $k \to \infty$, that is, asymptotically x_k becomes an approximate stationary point (and because of the descent condition this is an approximate minimiser),

- or else the angle θ_k converges to $\pi/2$, which is to say that the search direction asymptotically looses the property of being a descent direction.

Furthermore, if the objective function is bounded below. When this is not the case, the algorithm fails to terminate in finite time but produces a sequence $(x_k)_{\mathbb{N}}$ such that $\lim_{k\to\infty} f(x_k) = -\infty$, as is sensible.

Reading Assignment: Down-load and read Lecture-Note 2.