Numerical Optimization

Lecture Notes #21
Numerical Optimization — Summary and Guidelines

Peter Blomgren,
⟨blomgren.peter@gmail.com⟩

Department of Mathematics and Statistics
Dynamical Systems Group
Computational Sciences Research Center
San Diego State University
San Diego, CA 92182-7720

http://terminus.sdsu.edu/

Fall 2018

Outline

Recall: Our Problem is the "Easy Case"

We have been looking at local methods for the unconstrained, continuous, deterministic problem

$$\min_{\bar{\mathbf{x}} \in \mathbb{R}^n} f(\bar{\mathbf{x}})$$

where, most of the time, the objective $f(\bar{\mathbf{x}})$ is convex (or locally convexified).

Easier	**Harder**
Unconstrained	Constrained
Continuous Variables	Discrete Variables
Local Optimization	Global Optimization
Deterministic	Stochastic
Convex	Non-Convex

Table: Summary of some factors impacting the difficulty of the optimization problem.

Solving the Problem

We would like to find the **global minimizer**, *i.e.* a **solution** to the unconstrained optimization — a point $\bar{\mathbf{x}}^* \in \mathbb{R}^n$ such that

$$f(\bar{\mathbf{x}}^*) \leq f(\bar{\mathbf{x}},) \quad \forall \bar{\mathbf{x}} \in \mathbb{R}^n.$$

One of the first things we realized was that **in general we cannot achieve this goal.** Instead we solve **a sequence of local minimization problems**

$$\min_{\bar{\mathbf{x}} \in \mathbb{R}^n} m_k(\bar{\mathbf{x}}), \quad \text{where} \quad m_k(\bar{\mathbf{x}}) \approx f(\bar{\mathbf{x}}) \quad \text{for} \quad \bar{\mathbf{x}} \approx \bar{\mathbf{x}}_k.$$

Our goal is to generate a sequence of points $\{\bar{\mathbf{x}}_k\}_{k=0}^{\infty}$ which converge to a local minimum $\bar{\mathbf{x}}^*$, where $f(\bar{\mathbf{x}}^*) \leq f(\bar{\mathbf{x}})$, $\forall \bar{\mathbf{x}} \in N(\bar{\mathbf{x}}^*)$.

We have required our sequences to be **strictly monotone**, *i.e.*

$$f(\bar{\mathbf{x}}_0) > f(\bar{\mathbf{x}}_1) > \cdots > f(\bar{\mathbf{x}}_k) > f(\bar{\mathbf{x}}_{k+1}) > \cdots \geq f(\bar{\mathbf{x}}^*)$$

What We Have Studied

We have spent the last 20 lectures looking for **robust**, **efficient**, and **accurate** methods for

- finding the next iterate $\bar{\mathbf{x}}_{k+1}$,
- using information about the objective at the current point $\bar{\mathbf{x}}_k$.

In some cases — Conjugate Gradient (Truncated Newton) methods, and quasi-Newton methods — we also implicitly or explicitly use information about the objective at earlier iterates $\bar{\mathbf{x}}_j$, $j < k$.

We have looked at a significant number of methods; the purpose of this lecture is to review them and put them into a somewhat unified context.

Unconstrained Optimization: A Rough Roadmap

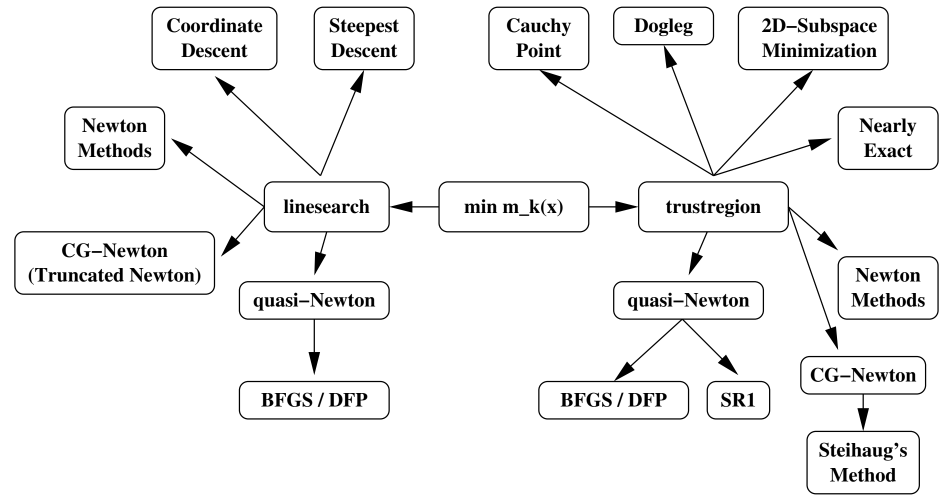

Figure: A very rough breakdown of the types of methods we have studied.

Line Search vs. Trust-Region

Line search based algorithms reduce the n-dimensional optimization problem to a one-dimensional problem

$$\min_{\bar{\mathbf{x}}\in\mathbb{R}^n} f(\bar{\mathbf{x}}) \quad \Rightarrow \quad \alpha_\mathbf{k} = \arg\min_{\alpha>\mathbf{0}} \mathbf{f}(\bar{\mathbf{x}}_\mathbf{k} + \alpha\bar{\mathbf{p}}_\mathbf{k}),$$

where $\bar{\mathbf{p}}_k$ is a chosen **search direction**; $\bar{\mathbf{x}}_{k+1} = \bar{\mathbf{x}}_k + \alpha_k\bar{\mathbf{p}}_k$.

Trust region based methods use a different approach. Using information gathered about the objective f, a simpler **model function** is generated.

A model function $m_k(\bar{\mathbf{x}})$ approximates the behavior of $f(\bar{\mathbf{x}})$ in a neighborhood of $\bar{\mathbf{x}}_k$, *e.g.* Taylor expansion

$$m_k(\bar{\mathbf{x}}_k + \bar{\mathbf{p}}) = f(\bar{\mathbf{x}}_k) + \bar{\mathbf{p}}^T\nabla f(\bar{\mathbf{x}}_k) + \frac{1}{2}\bar{\mathbf{p}}^T H_k\bar{\mathbf{p}}, \quad \text{where } H_k \approx \nabla^2 f(\bar{\mathbf{x}}_k).$$

Then the solution of the sub-problem gives $\bar{\mathbf{x}}_{k+1} = \bar{\mathbf{x}}_k + \bar{\mathbf{p}}_k$:

$$\bar{\mathbf{p}}_\mathbf{k} = \arg\min_{\bar{\mathbf{p}}\in\mathbf{N}(\bar{\mathbf{x}}_\mathbf{k})} \mathbf{m}_\mathbf{k}(\bar{\mathbf{x}}_\mathbf{k} + \bar{\mathbf{p}}), \quad \text{where } \mathbf{N}(\bar{\mathbf{x}}_\mathbf{k}) \text{ is the trust region.}$$

Line Search Methods: Subproblem #1 — The Search Direction

Line search methods break down into two major subproblems, which can be solved in a number of ways.

[1] **Identifying the search direction $\bar{\mathbf{p}}_k$.**

- [a] Steepest Descent: $\bar{\mathbf{p}}_k = -\nabla f(\bar{\mathbf{x}}_k)$.
- [b] Coordinate Descent: $\bar{\mathbf{p}}_k = -\bar{\mathbf{e}}_{[1+(k \bmod n)]}$.
- [c] "Pure" Newton Step: $\bar{\mathbf{p}}_k = -[\nabla^2 f(\bar{\mathbf{x}}_k)]^{-1}\nabla f(\bar{\mathbf{x}}_k)$.
- [d] Modified Newton Step: $\bar{\mathbf{p}}_k = -[\nabla^2 f(\bar{\mathbf{x}}_k) + E_k]^{-1}\nabla f(\bar{\mathbf{x}}_k)$.
- [c$_2$] Truncated (CG) Newton Step: $\bar{\mathbf{p}}_k \approx -[\nabla^2 f(\bar{\mathbf{x}}_k)]^{-1}\nabla f(\bar{\mathbf{x}}_k)$ (CG-solver for lin. sys.)
- [d$_2$] Truncated (CG) Modified Newton Step: $\bar{\mathbf{p}}_k \approx -[\nabla^2 f(\bar{\mathbf{x}}_k) + E_k]^{-1}\nabla f(\bar{\mathbf{x}}_k)$.
- [e] Quasi-Newton Step: $\bar{\mathbf{p}}_k = -B_k^{-1}\nabla f(\bar{\mathbf{x}}_k)$, where we have an update formula for B_k^{-1} — *e.g.* DFP, BFGS, or the restricted Broyden class.

Line Search Methods: Subproblem #2 — The Step Length

[2] **Identifying the step length α_k.**

Conditions needed to guarantee convergence:

[c_1] Wolfe Conditions — requires more than just descent; *i.e.* sufficient descent at each step.

[c_2] Strong Wolfe Conditions — slightly stronger than the Wolfe conditions; gives more descent than the Wolfe conditions in some cases.

[c_3] Goldstein conditions — similar to the Wolfe conditions. Often used in Newton-type methods, but not well suited for quasi-Newton methods.

Line Search Methods: Subproblem #2 — The Step Length

[2] Identifying the step length α_k.

Methods for finding α_k:

[m_1] Backtracking line-search (satisfies* the Wolfe conditions, without explicitly checking the second condition.)

[m_2] Line-search with quadratic / cubic interpolation.

The Initial Step Length $\alpha_k^{(0)}$:

[s_1] As we get close to the optimum $\bar{\mathbf{x}}^*$ we must test $\alpha_k^{(0)} = 1$ first. This is required in order to achieve maximal convergence rate for the overall method.

[s_2] Further away from $\bar{\mathbf{x}}^*$ we would like some clever heuristic so that, *e.g.* $\alpha_k^{(0)} \sim \alpha_{k-1}$.

Line Search Methods: Rules of Thumb

Since the cost of **quasi-Newton** methods (super-linear convergence) are essentially the same as the cost of **steepest descent** (linear convergence) — evaluation of the objective and its gradient — **steepest descent should never be used**.

Coordinate descent should only be used if evaluation of the gradient is very expensive, and the independent variables are loosely coupled. — The stronger the coupling of the variables, the worse the performance.

Newton methods (quadratic convergence) are preferred when evaluation of the Hessian $\nabla^2 f(\bar{\mathbf{x}})$ is reasonably cheap, and the solution of the linear system $\nabla^2 f(\bar{\mathbf{x}})\bar{\mathbf{p}}_k = -\nabla f(\bar{\mathbf{x}})$ can be computed efficiently. Otherwise **quasi-Newton** methods are preferred.

Line Search Methods: Rules of Thumb

The overall performance of **Newton methods** sometimes benefit greatly from an inexact solution of the linear system. A tolerance terminated CG-solution (truncated Newton) is a good choice, as long as the tolerance is chosen carefully in order to retain overall quadratic convergence.

The line search requires $\bar{\mathbf{p}}_k$ to be a descent direction, this cannot be guaranteed if $\nabla^2 f(\bar{\mathbf{x}})$ is indefinite. In order to guarantee convergence for Newton and truncated Newton methods, the Hessian must be modified $H_k = \nabla^2 f(\bar{\mathbf{x}}) + E_k$, so that H_k is Symmetric Positive Definite.

For **quasi-Newton** methods, BFGS is the preferred method in the line-search context.

Trust-Region Methods

The trust-region sub-problem

$$\bar{\mathbf{p}}_k = \arg\min_{\bar{\mathbf{p}} \in \mathbb{R}^n} m_k(\bar{\mathbf{p}}), \text{ such that } \bar{\mathbf{p}} \in T_k \overset{\text{usually}}{=} \{\bar{\mathbf{p}} \in \mathbb{R}^n : \|\bar{\mathbf{p}}\| \leq \Delta_k\},$$

where

$$m_k(\bar{\mathbf{p}}) = f(\bar{\mathbf{x}}_k) + \nabla f(\bar{\mathbf{x}}_k)^T \bar{\mathbf{p}} + \frac{1}{2}\bar{\mathbf{p}}^T B_k \bar{\mathbf{p}},$$

is a **locally constrained** minimization problem which (from a line-search-centric standpoint) gives both the direction and step length simultaneously.

There are several ways of approaching the (approximate, but sufficiently good) solution of the trust-region sub-problem.

Trust-Region Methods: The Sub-Problem

[1] The **Cauchy Point**, sufficient for global convergence

[a] The minimization of the quadratic model in the steepest descent direction. $\bar{\mathbf{p}}_k^s = \arg\min_{\bar{\mathbf{p}} \in T_k} f(\bar{\mathbf{x}}_k) + \nabla f(\bar{\mathbf{x}}_k)^T \bar{\mathbf{p}}$, then find $\tau_k = \arg\min_{\tau > 0} m_k(\tau \bar{\mathbf{p}}_k^s)$, such that $(\tau \bar{\mathbf{p}}_k^s) \in T_k$.

[2] Improvements to the Cauchy Point

[a] **Dogleg Method**: Minimize the objective over the path: $\bar{\mathbf{x}}_k \to \bar{\mathbf{p}}^U \to \bar{\mathbf{p}}^B$ subject to the trust-region constraint. Here $\bar{\mathbf{p}}^U$ is the unconstrained minimizer of the model in the steepest descent direction, and $\bar{\mathbf{p}}^B$ the **full step** $-B_k^{-1}\nabla f(\bar{\mathbf{x}}_k)$.

[b] **2D Subspace**: search for the minimizer of $m_k(\bar{\mathbf{p}})$ in the subspace (plane) spanned by the steepest descent direction and the full step, *i.e.*

$$\bar{\mathbf{p}}_k = \arg\min_{\bar{\mathbf{p}} \in \text{span}\{\nabla f(\bar{\mathbf{x}}_k),\, B_k^{-1}\nabla f(\bar{\mathbf{x}}_k)\}} m_k(\bar{\mathbf{p}})$$

Trust-Region Methods: The Sub-Problem

[2] Improvements to the Cauchy Point

[c] Note that the Cauchy point "lives" on the dogleg path, hence the dogleg method will do as well, or better than the Cauchy point. Further, the dogleg path is contained in the 2D subspace, hence the 2D-subspace minimization will do as well, or better than the dogleg method.

[d] For problems *with few independent variables*, the sub-problem can be solved **nearly exactly** by a Newton iteration on the parameter λ_k which makes $B_k + \lambda_k I$ symmetric positive semi-definite, and for which $\bar{\mathbf{p}}_k = [B_k + \lambda_k I]^{-1}\nabla f(\bar{\mathbf{x}}_k)$ either lies on the trust-region boundary, or $\lambda = 0$.

Trust-Region Methods: The Sub-Problem

[2] Improvements to the Cauchy Point

[e] If we relax the requirement on an exact solution of the simplified (dogleg, 2D subspace, nearly exact) subproblem, we can apply a truncated (CG) solver in the solution of the linear systems.

[3] The Hessian Approximation B_k

[a] If/when B_k is an "honest" attempt at approximating the Hessian $\nabla^2 f(\bar{\mathbf{x}}_k)$:

[α] The Cauchy point method should not be used — only linearly convergent.

[β] Dogleg OK if $\nabla^2 f(\bar{\mathbf{x}}_k)$ always is positive semidefinite.

[γ] When $\nabla^2 f(\bar{\mathbf{x}}_k)$ is indefinite, 2D-subspace (with Hessian modification $H_k = \nabla^2 f(\bar{\mathbf{x}}_k) + \lambda I$), nearly exact solution, or truncated Newton (Steihaug's method) works.

Trust-Region Methods: The Sub-Problem

[3] The Hessian Approximation B_k

[a$\beta\gamma$] If done correctly convergence rate is quadratic.

[b] B_k is updated by a quasi-Newton update-formula.

[α] Use BFGS update if it is known that the objective is convex, *i.e.* $\nabla^2 f(\bar{\mathbf{x}}_k)$ is SPD.

[β] Use SR1 update if $\nabla^2 f(\bar{\mathbf{x}}_k)$ is indefinite, since the SR1 update can capture this behavior.

[γ] Super-linear convergence-rate is expected of quasi-Newton methods.

Warning: In all cases (trust-region and line-search): Always be careful with **tolerances** and **bounds** in the hierarchy of sub-problems to be solved. One small mistake can easily break the expected quadratic or super-linear convergence, and give linear convergence (or worse).

1. Nonlinear Least Squares
2. Nonlinear Equations
3. Project Presentations

Index