An Overview of Optimization

Radoslav Harman

FMFI UK Bratislava

DATA SCIENCE

Two "components" of an optimization problem:

- X ... set of feasible (permissible) solutions
- $f : X \to \mathbb{R}$ ... objective (loss, error; utility, fitness) function, sometimes also "criterion"

A typical **minimization problem** is to find one element x^*, called **global optimum**, of:

$$\mathrm{argmin}_{x \in X} f(x) := \{x^* \in X : f(x^*) \leq f(x) \text{ for all } x \in X\}.$$

$$\text{Notation: } \min f(x) \text{ subject to (s.t.) } x \in X.$$

Variants: • Sometimes, we need to find more than one, possibly *all* elements of $\mathrm{argmin}_{x \in X} f(x)$. • More often, it is enough to find one or more $\tilde{x}$'s from X that are *"close"* to $\mathrm{argmin}_{x \in X} f(x)$.

Analogously we define a **maximization problem**.

Sometimes the objective function is $f : X \to \mathbb{R}^m$ for $m \geq 2$, in which case we get the so-called **multicriterial optimization** problem. This is a separate topic with specific principles.

Exercise: Assume that for each $x \in X$ we have a system of **neighbourhoods** $N_\epsilon(x) \subseteq X$; $\epsilon > 0$. How would you define the intuitive notion of a **"local" optimum**?

Exercise: Suggest optimization problems that may appear in practice, and the corresponding X and f. (Optimization of the movement of a robot, "training" in machine learning, ...)

Important questions:

- **Is there any solution?** ($\mathrm{argmin}_{x \in X} f(x) \neq \emptyset$?)
- **Is there a unique solution?** (Is $\mathrm{argmin}_{x \in X} f(x)$ a singleton?)
- **Can we verify the optimality** of a given $x^? \in X$?

Exercise: Give simple examples of optimization problems for which the solution does not exist, for which the solution exists and is unique and for which the solution does exist but is not unique.

Basic mathematical (analytic) conditions which guarantee the existence of the solution of an optimization problem:

- X is finite
- $X \subset \mathbb{R}^n$ **compact** & f **continuous** on X (in Eucl. metric).

Note: $X \subset \mathbb{R}^n$ is compact if it is closed and bounded.

Basic mathematical (analytic) condition that guarantees the uniqueness of the solution of a minimization problem:

- $X \subset \mathbb{R}^n$ is compact, **convex**, and f is **strictly convex** on X.

Note: $X \subset \mathbb{R}^n$ is convex and $f : X \in \mathbb{R}$ is strictly convex on X if for all $x_1, x_2 \in X$ and all $\alpha \in (0, 1)$ we have $\alpha x_1 + (1 - \alpha)x_2 \in X$ and

$$f(\alpha x_1 + (1 - \alpha)x_2) < \alpha f(x_1) + (1 - \alpha)f(x_2).$$

Some **classes of optimization problems** (loose definitions):

- X is finite or countable ... discrete optimization (combinatorial optimization, integer programming with $X \subseteq \mathbb{Z}^n$). Examples of applications: Optimum scheduling, Experimental design etc.
- $X \subseteq \mathbb{R}^n$ is not a discrete set ... continuous optimization. Examples of applications: Parameter estimation, "Training" of ANNs, etc.

—–

- $X \subseteq \mathbb{R}$ ... univariate optimization. Example of application: Line step search in multivariate optimization.
- $X \subseteq \mathbb{R}^n$, $n \geq 2$ ... multivariate optimization; almost all optimization problems in data science are multivariate.

—–

- $X = \mathbb{R}^n$ for some $n \geq 1$ ... unconstrained optimization
- $X \neq \mathbb{R}^n$ ... constrained optimization

- $X \subseteq \mathbb{R}^n$, f are convex ... convex optimization Applications: SVM, Some MLE, optimal approximate experimental design...
- Either X or f non-convex ... non-convex optimization; much of optimization in data science is non-convex

—–

- $X \subsetneq \mathbb{R}^n$ is polytopic, f is linear ... linear optimization (linear programming). Example of application: L1 regression.
- $X \subseteq \mathbb{R}^n$ is not polytopic or f is non-linear ... non-linear optimization (non-linear programming); most optimization problems in data science are non-linear.

—–

- f is differentiable on $X \subseteq \mathbb{R}^n$ ... differentiable optimization
- f is not differentiable in some point of X (in particular in the optimum) ... non-differentiable optimization

Basic **classes of methods** to solve optimization problems:

Finite discrete optimization:

- enumeration methods (brute-force complete, or "intelligent" incomplete methods such as the so-called branch-and-bound)
- greedy methods

Differentiable optimization:

- (quasi-) gradient methods (Newton's method, BFGS, ...)

Convex optimization:

- interior-point methods, specific methods for specific classes of convex programming (quadratic programming, SOCP, SDP, ...)
- simplex method for linear programming

Non-convex optimization and discrete optimization:

- various heuristic methods (hill-climbing, exchange methods, simulated annealing, genetic algorithms, particle swarm optimization, tabu search...) Many of them are *stochastic*, i.e., they use randomness in a fundamental way.

In R, we can use the following functions in the stats package:

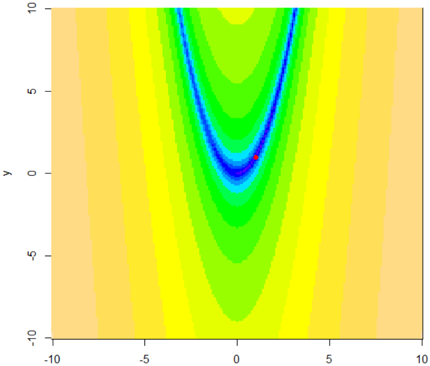
- optimize for one-dimensional, continuous, possibly non-differentiable optimization, either unconstrained, or constrained to an interval.
- optim for multi-dimensional, continuous optimization. Some implemented methods allow for non-differentiable optimization (for instance Nelder-Mead, SANN), some allow for differentiable optimization with box constraints (BFGS).
- nlm which implements a Newton-type optimization method for unconstrained, multivariate, differentiable optimization problem. See also nlminb.

Note: There are also other optimization methods in different R packages; they may be more appropriate or more efficient for the application at hand.

In statistics and, more generally, data science, we usually require multivariate continuous optimization (unconstrained or constrained), e.g., for the MLEs or for "training" of ML methods.

For benchmarking of multivariate continuous optimization algorithms: *Rosenbrock function* with $X \subseteq \mathbb{R}^2$ and the loss function

$$f(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2, \;\; (x_1, x_2)^T \in X.$$



Basic Optimization in R III

R Example Let us test various methods implemented in `optim` for finding the optimum of the Rosenbrock function.

- `BFGS` (Broyden-Fletcher-Goldfarb-Shanno)
- `CG` (Conjugate gradient method)
- `SANN` (simulated annealing; we will discuss it in detail later)
- `Nelder-Mead` (we will discuss it in detail later)

Let us also test the functions

- `nlm` from the standard package `stats`,
- `ga` from the contributed package `GA` (=genetic algorithms),
- `psoptim` from the contributed package pso (=particle swarm optimization).

That's all folks

See you in one week!