Optimization basics

Spring 2020

Amitabh Basu

Introduction to Data Science
Johns Hopkins University

The general optimization problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{such that} \quad & c_i(x) = 0 \quad i \in E \\ & c_i(x) \leq 0 \quad i \in I \end{aligned}$$

The general optimization problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{such that} \quad & c_i(x) = 0 \quad i \in E \\ & c_i(x) \leq 0 \quad i \in I \end{aligned}$$

- Unconstrained optimization: $E \cup I = \emptyset$
- Constrained optimization: $E \cup I \neq \emptyset$

The general optimization problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{such that} \quad & c_i(x) = 0 \quad i \in E \\ & c_i(x) \leq 0 \quad i \in I \end{aligned}$$

- Unconstrained optimization: $E \cup I = \emptyset$
- Constrained optimization: $E \cup I \neq \emptyset$

Will assume that f, $c_i, i \in E \cup I$ are all differentiable (even twice continuously differentiable) functions mapping $\mathbb{R}^n \to \mathbb{R}$

Fundamental notions from calculus

- For optimization theory and developing algorithms, we require tools for describing how function values change with their inputs.
- When derivatives exist, we use results from Calculus; e.g., gradients and Hessians

Fundamental notions from calculus

- For optimization theory and developing algorithms, we require tools for describing how function values change with their inputs.
- When derivatives exist, we use results from Calculus; e.g., gradients and Hessians

If $g : \mathbb{R}^n \to \mathbb{R}$ is differentiable, the gradient of g at x is

$$\nabla g(x) := \begin{pmatrix} \frac{\partial g(x)}{\partial x_1} \\ \vdots \\ \frac{\partial g(x)}{\partial x_n} \end{pmatrix}$$

Fundamental notions from calculus

If $g : \mathbb{R}^n \to \mathbb{R}$ is differentiable, the gradient of g at x is

$$\nabla g(x) := \begin{pmatrix} \frac{\partial g(x)}{\partial x_1} \\ \vdots \\ \frac{\partial g(x)}{\partial x_n} \end{pmatrix}$$

If $g : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable, the Hessian of g at x is

$$\nabla^2 g(x) := \begin{pmatrix} \frac{\partial^2 g(x)}{\partial x_1^2} & \cdots & \frac{\partial^2 g(x)}{\partial x_n \partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 g(x)}{\partial x_1 \partial x_n} & \cdots & \frac{\partial^2 g(x)}{\partial x_n^2} \end{pmatrix}$$

Fundamental notions from calculus

Vector valued functions:

If $F : \mathbb{R}^n \to \mathbb{R}^m$ is differentiable, the Jacobian of F at x is

$$J(x) := \nabla F(x) := \begin{pmatrix} \frac{\partial F_1(x)}{\partial x_1} & \cdots & \frac{\partial F_1(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_m(x)}{\partial x_1} & \cdots & \frac{\partial F_m(x)}{\partial x_n} \end{pmatrix}$$

where $F_i(x)$, $i = 1, \ldots, m$ is the i-th component of $F(x)$.

Fundamental notions from calculus

Vector valued functions:

If $F : \mathbb{R}^n \to \mathbb{R}^m$ is differentiable, the Jacobian of F at x is

$$J(x) := \nabla F(x) := \begin{pmatrix} \frac{\partial F_1(x)}{\partial x_1} & \cdots & \frac{\partial F_1(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_m(x)}{\partial x_1} & \cdots & \frac{\partial F_m(x)}{\partial x_n} \end{pmatrix}$$

where $F_i(x)$, $i = 1, \ldots, m$ is the i-th component of $F(x)$.

FACT: The Hessian is the Jacobian of the Gradient map.

Global and local solutions

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{such that} \quad & c_i(x) = 0 \quad i \in E \\ & c_i(x) \leq 0 \quad i \in I \end{aligned}$$

Define the feasible region as

$$\Omega := \left\{ x \in \mathbb{R}^n : \begin{array}{cccc} c_i(x) & = & 0 & i \in E, \\ c_i(x) & \leq & 0 & i \in I \end{array} \right\}$$

Global and local solutions

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{such that} \quad & c_i(x) = 0 \quad i \in E \\ & c_i(x) \leq 0 \quad i \in I \end{aligned}$$

Define the feasible region as

$$\Omega := \left\{ x \in \mathbb{R}^n : \begin{array}{ccc} c_i(x) & = & 0 \quad i \in E, \\ c_i(x) & \leq & 0 \quad i \in I \end{array} \right\}$$

$x^\star$ is a global optimum/minimizer if $f(x^\star) \leq f(x)$ for all $x \in \Omega$.

Global and local solutions

$$
\begin{aligned}
\min_{x \in \mathbb{R}^n} \quad & f(x) \\
\text{such that} \quad & c_i(x) = 0 \quad i \in E \\
& c_i(x) \leq 0 \quad i \in I
\end{aligned}
$$

Define the feasible region as

$$
\Omega := \left\{ x \in \mathbb{R}^n : \begin{array}{ccc} c_i(x) & = & 0 \quad i \in E, \\ c_i(x) & \leq & 0 \quad i \in I \end{array} \right\}
$$

$x^\star$ is a global optimum/minimizer if $f(x^\star) \leq f(x)$ for all $x \in \Omega$.

$x^\star$ is a local optimum/minimizer if there exists $\epsilon > 0$ such that $f(x^\star) \leq f(x)$ for all $x \in \Omega \cap B(x^\star, \epsilon)$, where $B(x^\star, \epsilon) := \{x \in \mathbb{R}^n : \|x^\star - x\| \leq \epsilon\}$.

Unconstrained optimization: optimality conditions

Unconstrained optimization: optimality conditions

We are interested in optimality conditions because they

- provide a means of guaranteeing when a candidate solution x is indeed optimal (sufficient conditions)
- indicate when a point is not optimal (necessary conditions)
- guide in the design of algorithms since

$$\text{lack of optimality} \quad \Leftrightarrow \quad \text{indication of improvement}$$

Unconstrained optimization: optimality conditions

First order necessary condition

THEOREM Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. If $x^\star$ is a local minimizer of f, then

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

Unconstrained optimization: optimality conditions

First order necessary condition

THEOREM Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. If $x^\star$ is a local minimizer of f, then

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

Second-order necessary conditions

THEOREM Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable. If $x^\star$ is a local minimizer of f, then $\nabla f(x^\star) = 0$ and $\nabla^2 f(x^\star)$ is positive semi-definite, i.e.,

$$s^T \nabla^2 f(x^\star) s \geq 0 \quad \text{for all } s \in \mathbb{R}^n$$

Unconstrained optimization: optimality conditions

Second-order sufficient conditions

THEOREM Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable. If $\nabla f(x^\star) = 0$ and $\nabla^2 f(x^\star)$ is positive definite, i.e.,

$$s^T \nabla^2 f(x^\star) s > 0 \quad \text{for all } s \neq 0,$$

then $x^\star$ is a local minimizer of f.

Unconstrained optimization: algorithms

Iterative numerical methods: generate iterates ("guesses") $x_0, x_1, \ldots$ such that these converge to a local minimizer, or at the very least to a stationary or critical point, i.e.,

$$\lim_{n\to\infty} x_n = x^\star, \qquad \text{and} \qquad \nabla f(x^\star) = 0.$$

Unconstrained optimization: algorithms

Iterative numerical methods: generate iterates ("guesses") $x_0, x_1, \ldots$ such that these converge to a local minimizer, or at the very least to a stationary or critical point, i.e.,

$$\lim_{n\to\infty} x_n = x^\star, \qquad \text{and} \qquad \nabla f(x^\star) = 0.$$

Two main paradigms:

1. Line Search Methods
2. Trust Region Methods

Unconstrained optimization: algorithms

Iterative numerical methods: generate iterates ("guesses") $x_0, x_1, \ldots$ such that these converge to a local minimizer, or at the very least to a stationary or critical point, i.e.,

$$\lim_{n\to\infty} x_n = x^\star, \qquad \text{and} \qquad \nabla f(x^\star) = 0.$$

Typical line search algorithm:

1. Initialize at x_0.
2. For $i = 0, 1, 2, \ldots$ until stopping criterion
   2.1 Choose a descent direction p_i such that $\nabla f^T(x_i) p_i < 0$.
   2.2 Do a line search on the one dimensional function $f(x_i + \eta p_i)$ to select step size $\eta_i \geq 0$
   2.3 Set $x_{i+1} := x_i + \eta_i p_i$

Unconstrained optimization: algorithms

Typical line search algorithm:

1. Initialize at x_0.
2. For $i = 0, 1, 2, \ldots$ until stopping criterion
   2.1 Choose a descent direction p_i such that $\nabla f^T(x_i)p_i < 0$.
   2.2 Do a line search on the one dimensional function $f(x_i + \eta p_i)$ to select step size $\eta_i \geq 0$
   2.3 Set $x_{i+1} := x_i + \eta_i p_i$

- Choice of stopping criterion: $\|\nabla f(x_i)\| \leq 10^{-6}\|\nabla f(x_0)\|$, Upper bound on number of iterations
- Choice of descent direction: $p_i = -\nabla f(x_i)$ (a.k.a. Gradient/Steepest descent), many other possibilities – modified Newton/quasi-Newton etc.
- Choice of step size:
  - Fixed, constant step size
  - Exact line search, i.e., find η_i that minimizes $f(x_i + \eta p_i)$,
  - Goldstein-Armijo condition: Set some global constant $0 < c < 1$ and select $\eta \geq 0$ such that

$$f(x_i + \eta p_i) \leq f(x_i) + \eta c \nabla f^T(x_i) p_i$$

Unconstrained optimization: algorithms

Sample convergence result:

THEOREM Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and ∇f is Lipschitz continuous on $\mathbb{R}^n$. Further assume that $f(x)$ is bounded from below on $\mathbb{R}^n$. Let $\{x_k\}$ be the iterates generated by the gradient descent algorithm with Goldstein-Armijo linesearch. Then there exists a constant M such that for all $T \geq 1$

$$\min_{k=0,\dots,T} \|\nabla f(x_k)\| \leq \frac{M}{\sqrt{T+1}}.$$

Consequently, for any $\epsilon > 0$, within $\left\lceil \left(\frac{M}{\epsilon}\right)^2 \right\rceil$ steps, we will see an iterate where the gradient has norm at most ϵ. In other words, we reach an "ϵ-stationary" point in $O((\frac{1}{\epsilon})^2)$ steps

Newton's method for root finding

Suppose we have a function

$$F : \mathbb{R}^n \to \mathbb{R}^n$$

and we wish to find $x^\star$ such that $F(x^\star) = 0$.

Newton's method for root finding

Suppose we have a function

$$F : \mathbb{R}^n \to \mathbb{R}^n$$

and we wish to find $x^\star$ such that $F(x^\star) = 0$.

Newton's method: Iteratively solve linear approximation.

Newton's method for root finding

Suppose we have a function

$$F : \mathbb{R}^n \to \mathbb{R}^n$$

and we wish to find $x^\star$ such that $F(x^\star) = 0$.

Newton's method: Iteratively solve linear approximation.

1. Initialize at x_0.
2. For $i = 0, 1, 2, \ldots$ until stopping criterion
   2.1 Solve the linear system $F(x_i) + \nabla F(x_i)(y - x_i) = 0$ for y.
   2.2 Set $x_{i+1} = y$, i.e.,

$$x_{i+1} = x_i - \nabla F(x_i)^{-1} F(x_i).$$

Newton's method for optimization

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

First order necessary condition:

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

So we are searching for zeros of the gradient map $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$.
Ready for applying Newton's method!

Initialize x^0 to some starting point. Compute iterates

$$x_{i+1} = x_i - (\nabla^2 f(x_i))^{-1} \nabla f(x_i)$$

Called the Newton step and $-(\nabla^2 f(x_i))^{-1} \nabla f(x_i)$ is called the Newton direction.

Newton's method for optimization

1. Hessian can be expensive to compute. Often simple approximations to Hessian are used: modified Newton, quasi-Newton.

2. Typically also combined with a line search: the full Newton step is not taken.

3. Better convergence guarantees because second order information is used.

Advantages of convexity

A function $f : \mathbb{R}^n \to \mathbb{R}$ is called convex if

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$$

for all $x, y \in \mathbb{R}^n$ and $\lambda \in [0,1]$.

Advantages of convexity

A function $f : \mathbb{R}^n \to \mathbb{R}$ is called convex if

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$$

for all $x, y \in \mathbb{R}^n$ and $\lambda \in [0,1]$.

Main advantages:

1. A local minimizer is a global minimizer!
2. Much better convergence rates for numerical optimization algorithms, e.g., $O\left(\frac{1}{\epsilon}\right)$ rate of convergence as opposed to $O\left(\left(\frac{1}{\epsilon}\right)^2\right)$ for gradient descent.
3. For constrained optimization, if the objective f and all the constraints c_i, $i \in E \cup I$ are convex, then it is called a convex optimization problem.

Constrained optimization: optimality conditions

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{such that} \quad & c_i(x) = 0 \quad i \in E \\ & c_i(x) \leq 0 \quad i \in I \end{aligned}$$

Constrained optimization: optimality conditions

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

$$\text{such that} \quad c_i(x) = 0 \quad i \in E$$

$$c_i(x) \le 0 \quad i \in I$$

First order necessary conditions: Karush-Kuhn-Tucker (KKT) conditions

THEOREM Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ and c_i, $i \in E \cup I$ are all continuously differentiable. If $x^\star$ is a local minimizer of f and some regularity conditions are satisfied, then there exist real scalars λ_i, $i \in E$ and $\mu_i \ge 0$, $i \in I$ such that

$$\begin{aligned}
\nabla f(x^\star) + \textstyle\sum_{i\in E} \lambda_i \nabla c_i(x^\star) + \sum_{i\in I} \mu_i \nabla c_i(x^\star) &= 0 \\
c_i(x^\star) &= 0 \;\; \forall i \in E \\
c_i(x^\star) &\le 0 \;\; \forall i \in I \\
\mu_i c_i(x^\star) &= 0 \;\; \forall i \in I
\end{aligned}$$

Constrained optimization: optimality conditions

First order necessary conditions: Karush-Kuhn-Tucker (KKT) conditions

THEOREM Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ and c_i, $i \in E \cup I$ are all continuously differentiable. If $x^\star$ is a local minimizer of f and some regularity conditions are satisfied, then there exist real scalars λ_i, $i \in E$ and $\mu_i \geq 0$, $i \in I$ such that

$$
\begin{array}{rcl}
\nabla f(x^\star) + \sum_{i\in E} \lambda_i \nabla c_i(x^\star) + \sum_{i \in I} \mu_i \nabla c_i(x^\star) & = & 0 \\
c_i(x^\star) & = & 0 \ \forall i \in E \\
c_i(x^\star) & \leq & 0 \ \forall i \in I \\
\mu_i c_i(x^\star) & = & 0 \ \forall i \in I
\end{array}
$$

Regularity conditions: Many different forms. Also called constraint qualification. Most common is the Linear Independence Constraint Qualification (LICQ):

Let $J(x^\star) \subseteq I$ index those inequality constraints that are satisfied at equality at $x^\star$, i.e., $c_i(x^\star) = 0$ for all $i \in J(x^\star)$. Then LICQ demands that $\nabla c_i(x^\star)$, $i \in J(x^\star)$ are linearly independent.

Algorithms: linear programming

Another regularity condition: the objective f and all constraints c_i, $i \in E \cup I$ are all affine functions (special case of convex), i.e.,

$$c_i(x) = \langle a_i, x \rangle + b_i, \quad i \in E \cup I$$

for some vectors $a_i \in \mathbb{R}^n$ and scalars $b_i \in \mathbb{R}$, and similarly

$$f(x) = \langle d, x \rangle$$

for some vector $d \in \mathbb{R}^n$.

This special case is called Linear Programming/Optimization.

Algorithms: linear programming

Another regularity condition: the objective f and all constraints c_i, $i \in E \cup I$ are all affine functions (special case of convex), i.e.,

$$c_i(x) = \langle a_i, x \rangle + b_i, \quad i \in E \cup I$$

for some vectors $a_i \in \mathbb{R}^n$ and scalars $b_i \in \mathbb{R}$, and similarly

$$f(x) = \langle d, x \rangle$$

for some vector $d \in \mathbb{R}^n$.

This special case is called Linear Programming/Optimization.

1. Most well studied optimization problem.
2. Main building block in more sophisticated algorithms.
3. KKT conditions are necessary and sufficient (assuming problem is feasible).
4. Highly efficient, specialized algorithms developed: Simplex method, Interior Point methods, Ellipsoid method.

Algorithms: general constraints and objective

$$
\begin{aligned}
\min_{x\in\mathbb{R}^n} \quad & f(x) \\
\text{such that} \quad & c_i(x) = 0 \quad i \in E \\
& c_i(x) \leq 0 \quad i \in I
\end{aligned}
$$

Algorithms: general constraints and objective

$$
\begin{aligned}
\min_{x \in \mathbb{R}^n} \quad & f(x) \\
\text{such that} \quad & c_i(x) = 0 \quad i \in E \\
& c_i(x) \leq 0 \quad i \in I
\end{aligned}
$$

- Penalty Methods. Reduce to an unconstrained problem: Choose some $\gamma > 0$ and minimize the function

$$\phi(x) := f(x) + \gamma \sum_{i \in E} c_i(x)^2 + \gamma \sum_{i \in I} (\min 0, c_i(x))^2$$

- Augmented Lagrangian Methods.
- Interior Point Methods. Adapt Newton's method to the constrained setting.

References

General nonlinear, nonconvex optimization:

1. **Numerical Optimization**, Jorge Nocedal and Stephen J. Wright, Springer 2006.
2. **Optimization methods for large-scale machine learning**, Léon Bottou, Frank E Curtis, Jorge Nocedal, SIAM Review, vol. 60 (2), pp. 223-311, 2018.
3. **Linear and nonlinear programming**, David G. Luenberger and Yinyu Ye. Addison-Wesley, 1984.

Convex optimization:

1. **Convex Optimization**, Stephen Boyd and Lieven Vandenberghe, Cambridge University Press, 2004.
2. **Introductory Lectures on Convex Optimization**, Yuri Nesterov, Kluwer Academic Publishers, 2004.

Linear programming/optimization:

1. **Linear Programming**, Vasek Chvatal, W.H. Freeman and Company, 1983.
2. **Primal-Dual Interior Point Methods**, Stephen J. Wright, SIAM, 1997.

Software

1. SciPy (nonconvex, nonlinear)
2. NITRO (nonconvex, nonlinear)
3. IPOPT (nonconvex, nonlinear)
4. CVX (convex)
5. GUROBI (convex and linear)
6. CPLEX (convex and linear)
7. BARON (global optimization)
8. SCIP (global optimization)
9. Couenne (global optimization)