Optimization Intro

Stephen J. Wright[1]

[2]Computer Sciences Department,
University of Wisconsin-Madison.

IMA, August 2016

Course Materials

The site `www.cs.wisc.edu/~swright/nd2016/` contains

- slides
- notes and background material
- exercises.

What is Optimization?

Make the optimal choice from among a number of alternative possibilities.

- The "number" is large, often infinite.
- Need a mathematical definition of "best": objective function.
- Need a mathematical way to define the "choices": variables.
- Need a way to describe restrictions on choices: constraints.

Variables, Objective, Constraints are the key ingredients of all optimization problems.

Formulation of an application as an mathematical optimization problem is often nontrivial. Required interaction with applications experts, "domain scientists."

Sometimes the formulation process alone leads to new research areas (e.g. compressed sensing).

Optimization

The field of Optimization conventionally encompasses

- analyzing fundamental properties of the mathematical formulation;
- devising algorithms to solve optimization formulations, and analyzing the mathematical properties of these algorithms (convergence, complexity, efficiency);
- Also implementation and testing of the algorithms.

We'll discuss mostly the second topic in this course, referring to the first for the necessary foundations (e.g. recognizing solutions).

Standard mathematical formulations are the interface between applications and optimization. Paradigms such as linear programming, quadratic programming, semidefinite programming are rich areas of mathematical research and also extremely useful in applications.

But....

New Developments

But increasingly, optimization is about direct engagement with the applications community. It's very important to understand the context in which the problems arise, and what the user needs from their use of optimization technology.

Formulations and algorithms must be customized to new and important applications and their contexts. To be useful to the user, the optimization technology must be cognizant of these features — shoehorning it into one of the traditional paradigms is often not enough. We see this in data analysis and machine learning, for instance.

New, important applications are driving new, fundamental research in optimization.

Solvers must be assembled from various formulation and algorithmic tools from the "optimization toolbox."

A thorough knowledge of the toolbox is a great skill to have in the current environment!

Variables and Constraints

Usually consider variables to be (column) vectors of real numbers: $x = (x_1, x_2, \ldots, x_n)^T$.

Matrices (as data or variables) usually denoted by Roman caps: A, B, X, etc.

(In some problems, the variables are *functions*, but if these are parametrized, the problem reduces to optimization over a real vector.)

Constraints on variables can be expressed geometrically, e.g.

$$x \in \Omega, \quad \text{where } \Omega \text{ is a closed convex set.}$$

Also algebraically. There are linear constraints:

$$Ax \geq b, \quad Cx = d,$$

where A and C are matrices and b and d are vectors.

$A \geq b$ is an intersection of half-spaces and $Cx = d$ is an affine space (which is parallel to a subspace in $\mathbb{R}^n$).

Constraints

The feasible set is the set in $\mathbb{R}^n$ consisting of points that satisfy all the constraints.

Note that the feasible set is an intersection of the sets defined by all the constraints — not a union.

(Unions of different constraint sets are disjunctions. It is possible to handle these but the optimization problem becomes much harder, so we punt that topic to next week!)

Nonlinear constraints can be expressed as

$$c_i(x) \leq 0, \quad \text{where } c_i : \mathbb{R}^n \to \mathbb{R}, \quad i \in \mathcal{I}.$$

Each function typically defines a manifold in $\mathbb{R}^m$. But in generally it could be much more complicated e.g. $\cos x \leq 1/2$ defines a set with many pieces.

Can assemble these into a vector: $c_I(x) \leq 0$, where $c_I = (c_i)_{i \in \mathcal{I}}$.

Constraints

Note that the inequalities are non-strict. In general, a strict inequality implies an open set, and optimization problems often do not attain their minimizers over open sets. e.g.

$$\min x \ \text{ s.t. } x > 0 \quad \text{has no solution.}$$

We can have nonlinear equality constraints too: $c_E(x) = 0$, where $c_E = (c_i)_{i \in \mathcal{E}}$.

A special type of linear constraint is a bound constraint which has the form $x_i \geq l_i$ and/or $x_i \leq u_i$, where x_i is the i-th element of x and l_i and u_i are scalars too. These are linear constraints that depend on just a single component of x, and algorithms can usually exploit their simplicity.

Can assemble bounds into vectors, and write as

$$l \leq x \leq u.$$

Nonnegativity bounds often appear: $x \geq 0$.

Semidefiniteness

If the variable is a symmetric $n \times n$ matrix X, we typically have the constraint in semidefinite programming that X is positive semidefinite, denoted by $X \succeq 0$.

This constraint can be expressed by n nonlinear inequality constraints (i.e. the determinants of the leading minors are all nonnegative) but this is a messy way to do it, involving too many nonlinear constraints. There is a much more graceful theory that handles such constraints directly.

Elementary Topology

Subsets of $\mathbb{R}^n$. Look to Wikipedia for definitions! Also see some notes posted on the site `www.cs.wisc.edu/~swright/nd2016/`

- (Open) neighborhood of a point x.
- Open set
- Closed set
- Compact set
  - Every cover has a finite subcover
  - Every sequence $\{x^k\}$ in this set has an accumulation point in the set.
- Heine-Borel Theorem: "Closed and Bounded = Compact"
- Bolzano-Weierstrass Theorem: "Every bounded sequence in $\mathbb{R}^n$ has a convergent subsequence."

Linear Algebra

Numerical linear algebra plays a central role in numerical optimization.

Subproblems in optimization algorithms are frequently numerical linear algebra problems. Need to solve efficiently – exploit structure.

Also, the linear algebra problems at adjacent iterations of the optimization algorithm may be closely related, especially in the simplex method for linear programming. Thus need to do specialized linear operations, particularly factorization updates efficiently and stably.

Rank of a matrix A is the dimension of the subspace spanned by the rows of A (which is the same as dimension of the subspace spanned by the columns of A).

Key Factorizations

Factorizations of matrices are needed to solve linear equations stably. Factorization obviates need for explicitly calculating inverses and are more efficient and more stable (if appropriate precautions are taken).

A square and symmetric positive definite: Cholesky factorization:

$$A = LL^T, \quad \text{where } L \text{ is lower triangular}$$

A general rectangular, $m \times n$:

$$A = QR, \quad \text{where } Q \text{ is orthogonal and } R \text{ is upper triangular.}$$

$$A = LU, \quad \text{where } L \text{ is lower triangular and } U \text{ is upper triangular.}$$

Also the singular value decomposition (SVD):

$$A = USV^T,$$

where U and V have orthonormal columns and S is a diagonal matrix with nonnegative diagonals $(\sigma_1, \sigma_2, \ldots, \sigma_r)$, where $r =$ rank of A.

Permutations are Important!

Cholesky and SVD don't require the order of rows or columns of A to be permuted. (In SVD the permutations are incorporated into U and V.) But the LU and QR factorizations may require this. Thus may need to find permutation matrices P, $\bar{P}$ on the fly, to maintain stability of factorizations:

$$PA\bar{P}^T = LU, \quad A\bar{P}^T = QR,$$

where P and $\bar{P}$ are permutation matrices.

Given factorizations, it's easy to solve linear equations and linear least sq.

A square and nonsingular: Can write $Ax = b$ equivalently as

$$PA\bar{P}^T(\bar{P}x) = Pb \quad \Leftrightarrow \quad LU(\bar{P}x) = Pb.$$

Then solve by doing triangular substitutions with L and U:

$$\text{solve } Lz = Pb; \quad \text{solve } Uy = z; \quad \text{set } x = \bar{P}^T y.$$

Exercise: Solve $\min_x \|Ax - b\|_2^2$ using QR factorization or SVD.

Sequences, Limits, Accumulation Points

Consider sequences $\{x^k\}_{k=1,2,\ldots}$ with $x^k \in \mathbb{R}^n$, $\{\alpha_k\}_{k=1,2,\ldots}$ with $\alpha_k \in \mathbb{R}_+$.

- limit: $\lim x^k = \bar{x}$ if for all $\epsilon > 0$, have $\|x^k - \bar{x}\| \leq \epsilon$ for all k suff large.
- $\bar{x}$ is accumulation point: if for all $\epsilon > 0$ and all K, there is $k > K$ with $\|x^k - \bar{x}\| \leq \epsilon$.

For scalar sequences:

- infimum: largest value of $\bar{\alpha}$ such that $\alpha_k \geq \bar{\alpha}$ for all k.
- supremum: smallest value of $\bar{\alpha}$ such that $\alpha_k \leq \bar{\alpha}$ for all k.
- lim inf: is $\lim_{K\to\infty}(\inf_{k\geq K} \alpha_k)$. Example: $\lim\inf -2^{-k} = 0$.
- lim sup: is $\lim_{K\to\infty}(\sup_{k\geq K} \alpha_k)$. Example: $\lim\sup(-1)^k + 2^{-k} = 1$.

Rates of Convergence for scalar sequence $\{\alpha_k\}$

- Q-linear (geometric): There is $\rho \in (0,1)$ such that

$$\frac{\alpha_{k+1}}{\alpha_k} \leq \rho, \quad k = 1, 2, \ldots .$$

- Sublinear: A variety of arithmetic rates. For some C, have

$$\alpha_k \leq \frac{C}{\sqrt{k}} \text{ or } \alpha_k \leq \frac{C}{k} \text{ or } \alpha_k \leq \frac{C}{k^2}.$$

- Q-superlinear:

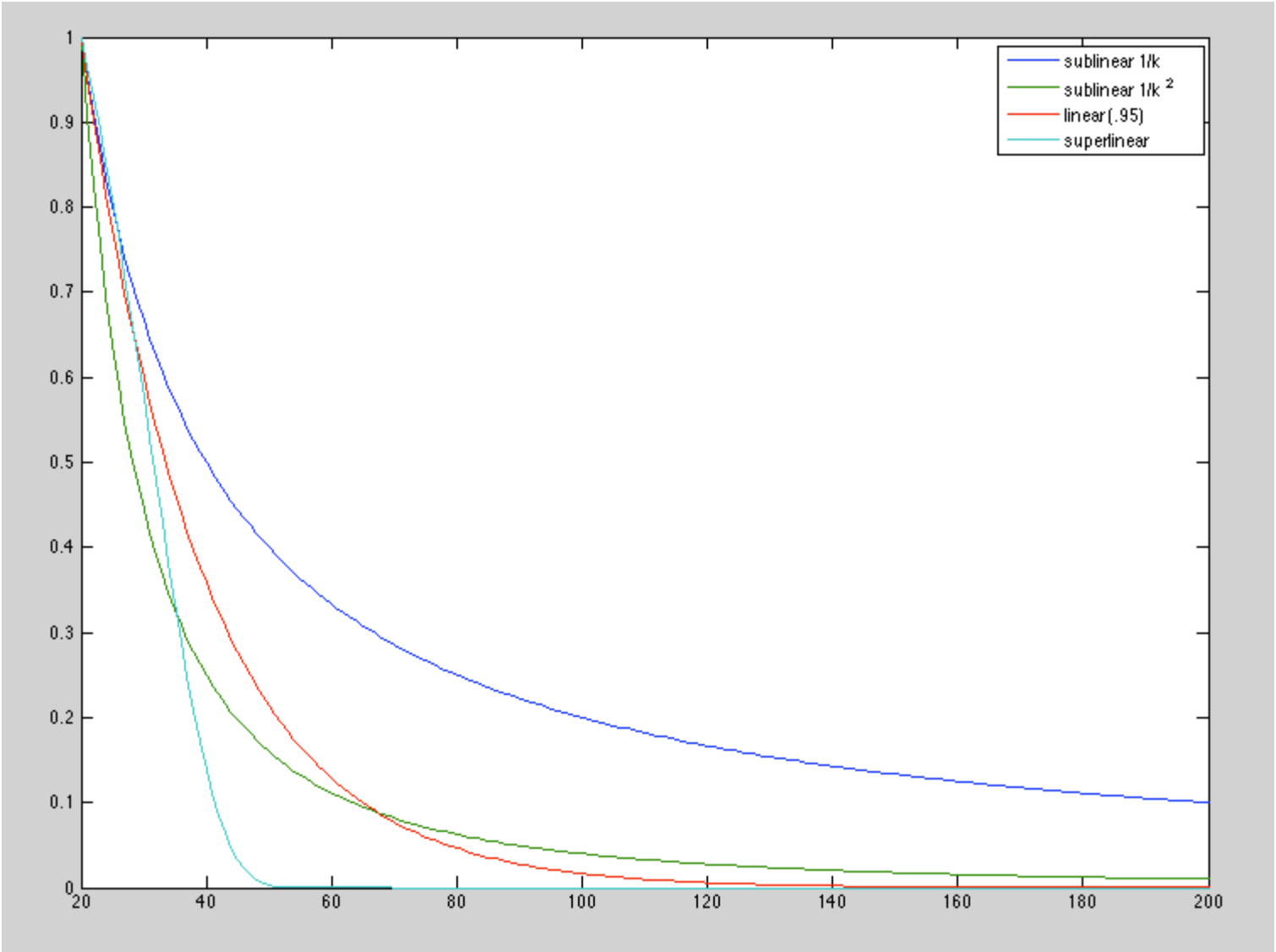
$$\frac{\alpha_{k+1}}{\alpha_k} \to 0 \text{ as } k \to \infty.$$

- Q-quadratic: There is C such that

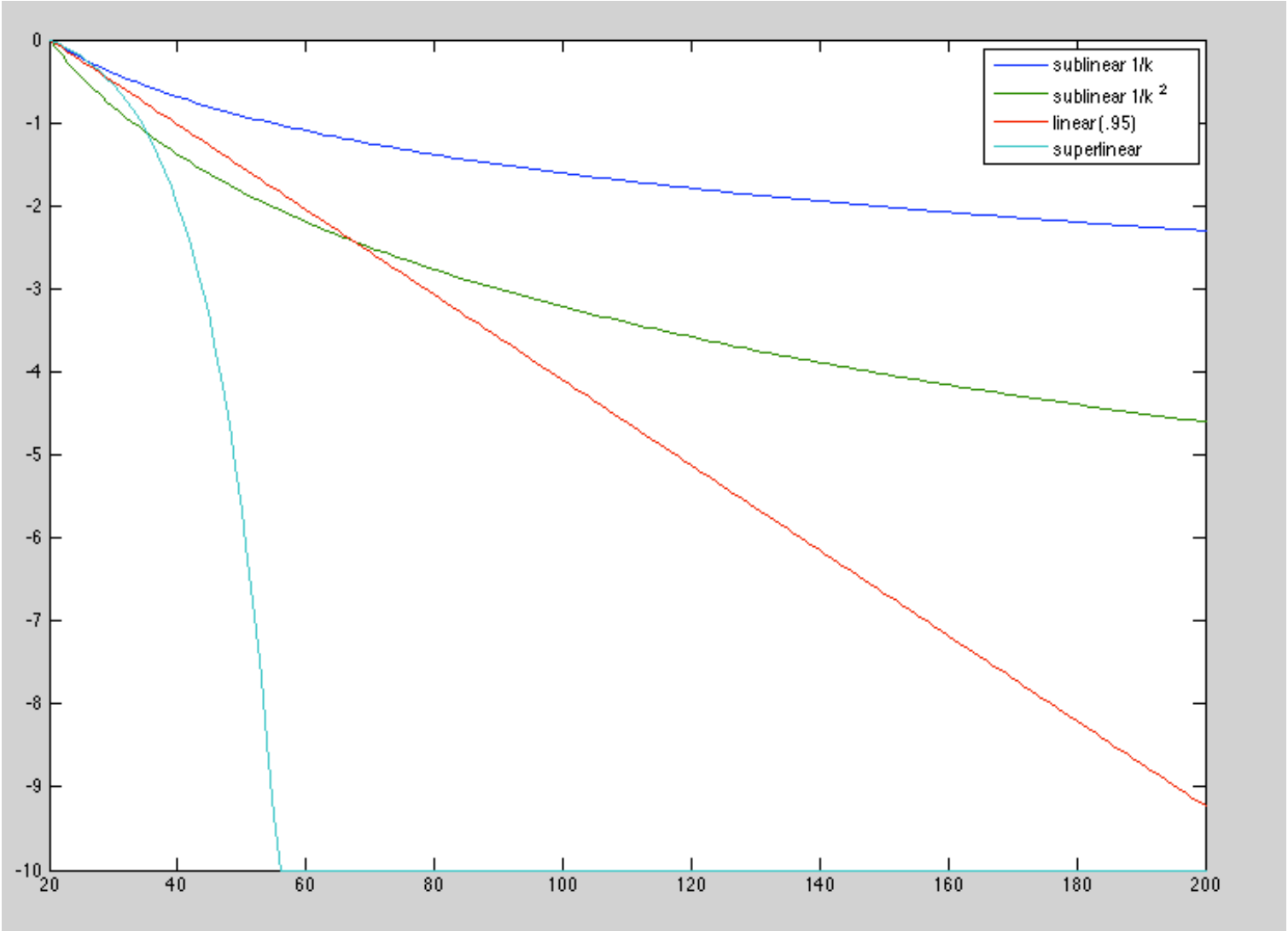
$$\frac{\alpha_{k+1}}{\alpha_k^2} \leq C.$$

R-linear, R-superlinear, R-quadratic are when there is another sequence $\{\gamma_k\}$ with $0 \leq \alpha_k \leq \gamma_k$ for all k, and $\{\gamma_k\}$ is Q-linear, Q-superlinear, Q-quadratic. Examples! See pp. 619-620 in Nocedal and Wright (2006).

Convergence Rates: Standard Plots

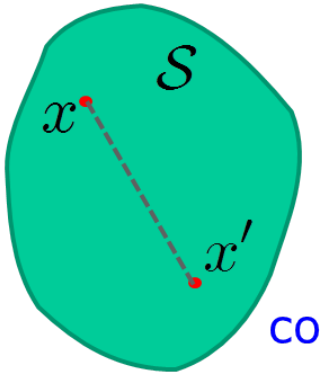


Convergence Rates: Log Plots

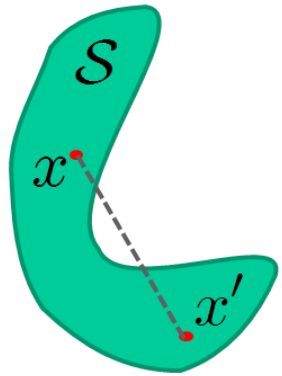


Convex and strictly convex sets

$\mathcal{S}$ is convex if $x, x' \in \mathcal{S} \Rightarrow \forall \lambda \in [0,1], \quad \lambda x + (1-\lambda)x' \in \mathcal{S}$

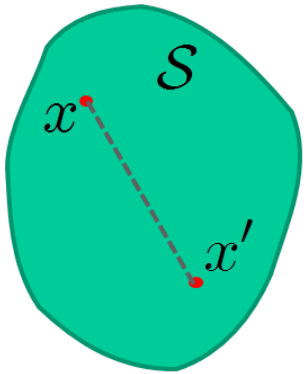


convex

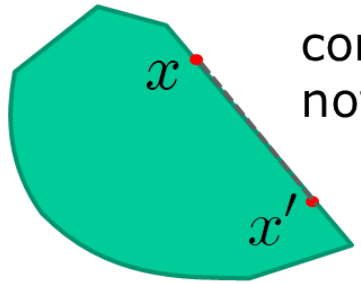


non-convex

$\mathcal{S}$ is strictly convex if $x, x' \in \mathcal{S} \Rightarrow \forall \lambda \in (0,1), \quad \lambda x + (1-\lambda)x' \in \text{int}(\mathcal{S})$



strictly convex



convex, but not strictly

Review of Basics: Convex Functions

Extended real valued function: $f : \mathbb{R}^N \to \bar{\mathbb{R}} = \mathbb{R} \cup \{+\infty\}$

Domain: $\mathrm{dom}(f) = \{x : f(x) \neq +\infty\}$

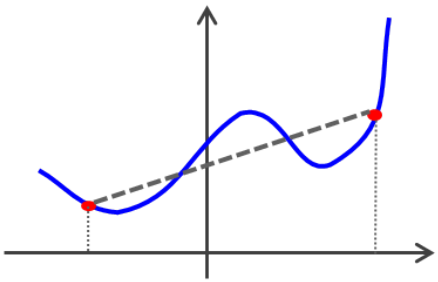
f is proper if $\mathrm{dom}(f) \neq \emptyset$

f is convex if

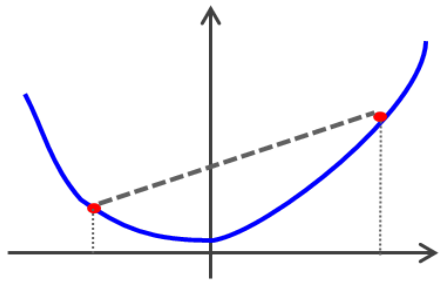
$$\forall \lambda \in [0,1], x, x' \in \mathrm{dom}(f) \quad f(\lambda x + (1-\lambda)x') \leq \lambda f(x) + (1-\lambda) f(x')$$

f is strictly convex if

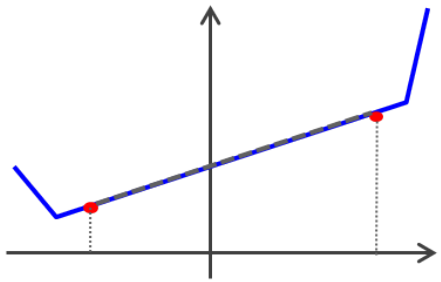
$$\forall \lambda \in (0,1), x, x' \in \mathrm{dom}(f) \quad f(\lambda x + (1-\lambda)x') < \lambda f(x) + (1-\lambda) f(x')$$



non-convex



strictly convex



convex, not strictly

Taylor's Theorem

Taylor's theorem is a central tool in smooth nonlinear optimization. It shows how a function value $f(y)$ can be approximated in terms of the values of f and its derivatives at a nearby point x.

Instead of minimizing f, we can replace f with a simplified approximation based on the Taylor-series approximation.

Doing this repeatedly leads to an iterative algorithm

See (Nocedal and Wright, 2006, p. 15), and also Chapter 1 in the notes distributed.

Taylor's Theorem

Theorem

Given a continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, *and given* $x, p \in \mathbb{R}^n$, *we have that*

$$f(x+p) = f(x) + \int_0^1 \nabla f(x+\gamma p)^T p \, d\gamma,$$

$$f(x+p) = f(x) + \nabla f(x+\gamma p)^T p, \quad \text{some } \gamma \in (0,1).$$

If f *is twice continuously differentiable, we have*

$$\nabla f(x+p) = \nabla f(x) + \int_0^1 \nabla^2 f(x+\gamma p) p \, d\gamma,$$

$$f(x+p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x+\gamma p) p, \quad \text{some } \gamma \in (0,1).$$

Fun with Taylor's Theorem

Many useful results can be derived from Taylor's theorem that are useful in the design and analysis of algorithms.

If L is the Lipschitz constant for ∇f, that is,

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad \text{for all } x, y,$$

then if f is convex, we have

$$\frac{\mu}{2}\|y - x\|^2 \leq f(y) - f(x) - \nabla f(x)^T(y - x) \leq \frac{L}{2}\|y - x\|^2,$$

for some $\mu \geq 0$. μ is called the modulus of convexity for f.

We also have

$$f(x) + \nabla f(x)^T(y - x) + \frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|^2 \leq f(y), \tag{1}$$

$$\frac{1}{L}\|\nabla f(x) - \nabla f(y)\|^2 \leq (\nabla f(x) - \nabla f(y))^T(x - y) \leq L\|x - y\|^2. \tag{2}$$

If, in addition, f is strongly convex with modulus μ and unique minimizer x^*, we have for all $x, y \in \text{dom}(f)$ that

$$f(y) - f(x) \geq -\frac{1}{2\mu}\|\nabla f(x)\|^2. \tag{3}$$

Local and Global Solutions

Consider the general problem

$$\min f(x) \text{ s.t. } x \in \Omega,$$

where Ω is a closed set.

x^* is a global solution if

$$x^* \in \Omega, \quad f(x^*) \leq f(x) \text{ for all } x \in \Omega.$$

Global solutions are hard to find in general, unless f and Ω are convex.

x^* is a local solution if $x^* \in \Omega$ and there is a neighborhood $\mathcal{N}$ of x^* such that

$$f(x^*) \leq f(x) \text{ for all } x \in \mathcal{N} \cap \Omega.$$

x^* is a strict local solution if $x^* \in \Omega$ and there is a neighborhood $\mathcal{N}$ of x^* such that

$$f(x^*) < f(x) \text{ for all } x \in \mathcal{N} \cap \Omega \text{ with } x \neq x^*.$$

x^* is an isolated local solution if there is a neighborhood of x^* containing no other local solutions.

Convex Sets: Normal Cones

If Ω is a closed convex set, the *normal cone* $N_\Omega(x)$ is:

$$N_\Omega(x) = \{d \in \mathbb{R}^n \ : \ d^T(y - x) \leq 0 \ \text{ for all } y \in \Omega\}.$$

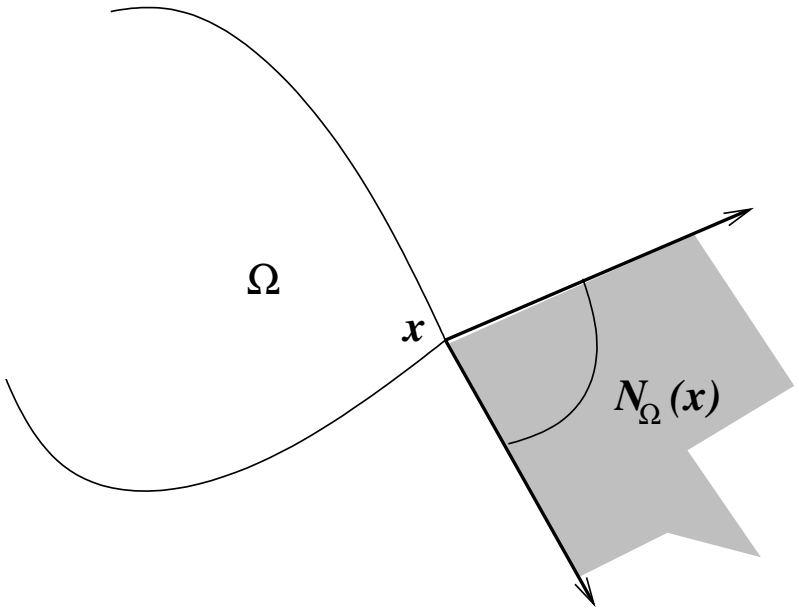


Figure: Normal Cone

(Some other examples....)

Projection onto Convex Sets

If Ω is convex, projection operator is:

$$P_\Omega(x) = \arg\min_{z \in \Omega} \|z - x\|.$$

Lemma

(i) $(P(y) - z)^T (y - z) \geq 0$ *for all* $z \in \Omega$, *with equality if and only if* $z = P(y)$.

(ii) $(y - P(y))^T (z - P(y)) \leq 0$ *for all* $z \in \Omega$.

Proof of (i).

For any $z \in \Omega$, have

$$\begin{aligned} \|P(y) - y\|_2^2 &= \|P(y) - z + z - y\|_2^2 \\ &= \|P(y) - z\|_2^2 + 2(P(y) - z)^T(z - y) + \|z - y\|_2^2 \end{aligned}$$

so by rearrangement

$$2(P(y) - z)^T(y - z) = \|P(y) - z\|_2^2 + \left[\|z - y\|_2^2 - \|P(y) - y\|_2^2\right]. \quad (4)$$

The term in [] is nonnegative, from the definition of P. Thus have ≥ 0.

If $z = P(y)$, obviously $(P(y) - z)^T(y - z) = 0$. If the latter condition holds, then first term in (4) is zero, so have $z = P(y)$. □

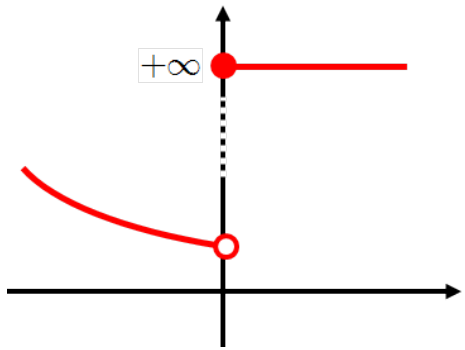
Lower Semi-Continuity: Why Is It Important?

A function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ is lower semi-continuous (l.s.c.) if

$$\liminf_{x \to x_0} f(x) \geq f(x_0), \quad \text{for any } x_0 \in \text{dom}(f)$$

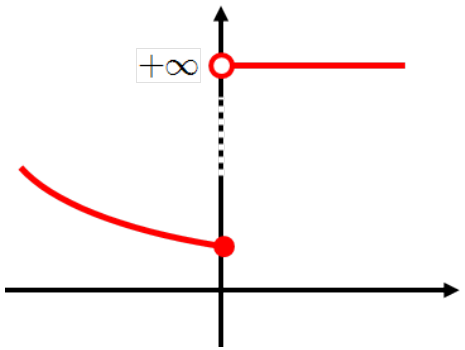
or, equivalently, $\{x : f(x) \leq \alpha\}$ is a closed set, for any $\alpha \in \mathbb{R}$

$$f(x) = \begin{cases} e^{-x}, & \text{if } x < 0 \\ +\infty, & \text{if } x \geq 0 \end{cases}$$



$\text{dom}(f) =]-\infty, 0[, \quad \arg\min_x f(x) = \emptyset$

$$f(x) = \begin{cases} e^{-x}, & \text{if } x \leq 0 \\ +\infty, & \text{if } x > 0 \end{cases}$$



$\text{dom}(f) =]-\infty, 0], \quad \arg\min_x f(x) = \{0\}$

Unless stated otherwise, we only consider l.s.c. functions.

Coercivity, Convexity, and Minima

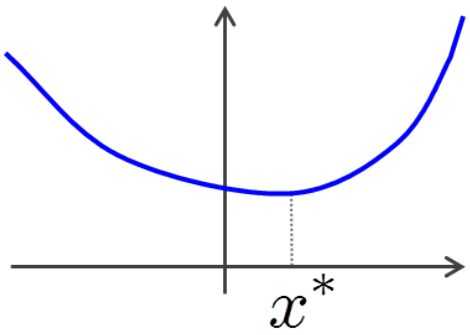
$$f : \mathbb{R}^N \to \bar{\mathbb{R}} = \mathbb{R} \cup \{+\infty\}$$

f is coercive if $\lim_{\|x\| \to +\infty} f(x) = +\infty$

if f is coercive, then $G \equiv \arg\min_x f(x)$ is a non-empty set

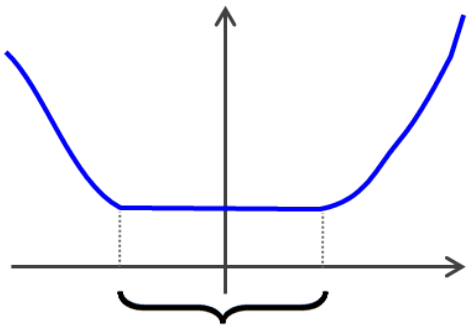
if f is strictly convex, then G has at most one element

coercive and strictly convex



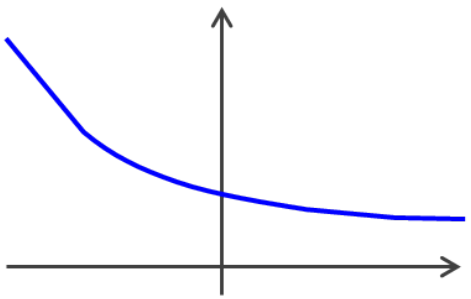
$G = \{x^*\}$

coercive, not strictly convex



G

convex, not coercive



$G = \emptyset$

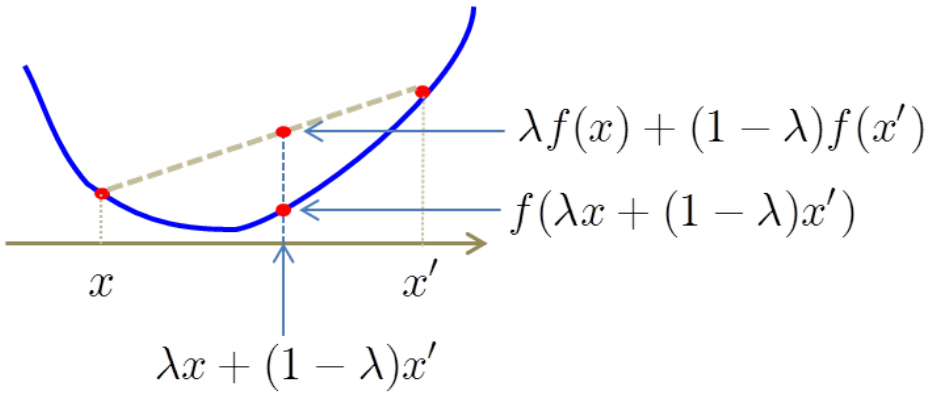
Another Important Concept: Strong Convexity

Recall the definition of convex function: $\forall \lambda \in [0,1]$,

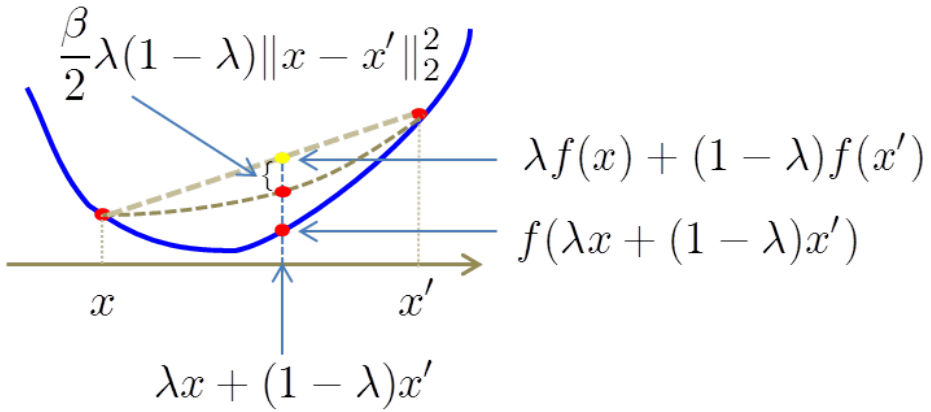
$$f(\lambda x + (1-\lambda)x') \leq \lambda f(x) + (1-\lambda) f(x')$$

A $\beta-$strongly convex function satisfies a stronger condition: $\forall \lambda \in [0,1]$

$$f(\lambda x + (1-\lambda)x') \leq \lambda f(x) + (1-\lambda) f(x') - \frac{\beta}{2}\lambda(1-\lambda)\|x - x'\|_2^2$$



convexity



strong convexity

Strong convexity $\Rightarrow$ strict convexity.

$\not\Leftarrow$

A Little More on Convex Functions

Let $f_1, ..., f_N : \mathbb{R}^n \to \bar{\mathbb{R}}$ be convex functions. Then

- $f : \mathbb{R}^n \to \bar{\mathbb{R}}$, defined as $f(x) = \max\{f_1(x), ..., f_N(x)\}$, is convex.
- $g : \mathbb{R}^n \to \bar{\mathbb{R}}$, defined as $g(x) = f_1(L(x))$, where L is affine, is convex.
  Note: L is affine $\Leftrightarrow$ $L(x) - L(0)$ is linear; e.g. $L(x) = Ax + b$.
- $h : \mathbb{R}^n \to \bar{\mathbb{R}}$, defined as $h(x) = \sum_{j=1}^{N} \alpha_j f_j(x)$, for $\alpha_j > 0$, is convex.

An important function: the indicator of a set $C \subset \mathbb{R}^n$,

$$\iota_C : \mathbb{R}^n \to \bar{\mathbb{R}}, \quad \iota_C(x) = \begin{cases} 0 & \Leftarrow \quad x \in C \\ +\infty & \Leftarrow \quad x \notin C \end{cases}$$

If C is a closed convex set, ι_C is a l.s.c. convex function.

Smooth Convex Functions

Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice differentiable and consider its Hessian matrix at x, denoted $\nabla^2 f(x)$ (or $Hf(x)$):

$$\left(\nabla^2 f(x)\right)_{ij} = \frac{\partial f}{\partial x_i \partial x_j}, \quad \text{for } i, j = 1, \ldots, n.$$

- f is convex $\Leftrightarrow$ its Hessian $\nabla^2 f(x)$ is positive semidefinite $\forall_x$
- f is strictly convex $\Leftarrow$ its Hessian $\nabla^2 f(x)$ is positive definite $\forall_x$
- f is β-strongly convex $\Leftrightarrow$ its Hessian $\nabla^2 f(x) \succeq \beta I$, with $\beta > 0$, $\forall_x$.

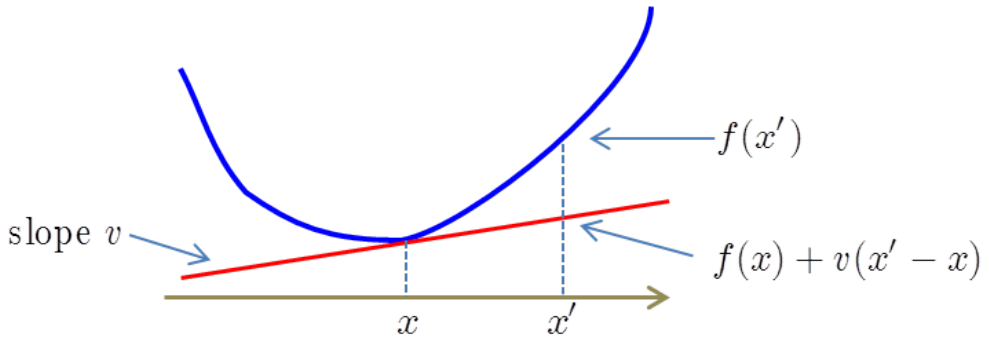
Subgradients

Subgradients generalize gradients for general convex functions:

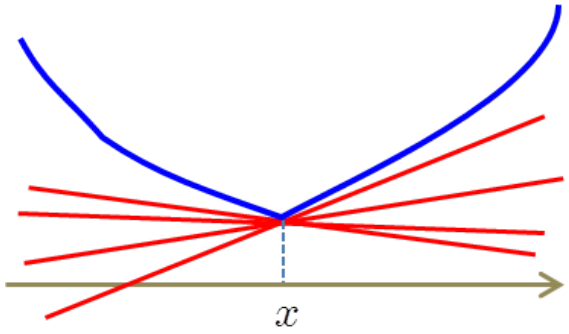
$$\boxed{v \text{ is a subgradient of } f \text{ at } x \text{ if } f(x') \geq f(x) + v^T(x' - x)}$$

Subdifferential: $\partial f(x) = \{\text{all subgradients of } f \text{ at } x\}$

If f is differentiable, $\partial f(x) = \{\nabla f(x)\}$



linear lower bound



nondifferentiable case

Subgradients satisfy a monotonicity property: If $a \in \partial f(x)$ and $b \in \partial f(y)$, then $(a - b)^T(x - y) \geq 0$.

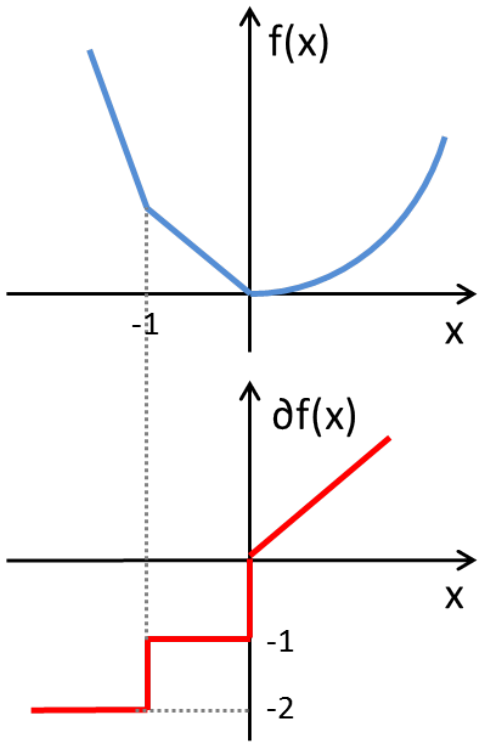
More on Subgradients and Subdifferentials

The subdifferential is a set-valued function:

$$f : \mathbb{R}^d \to \mathbb{R} \quad \Rightarrow \quad \partial f : \mathbb{R}^d \to \text{subsets of } \mathbb{R}^d$$

$$f(x) = \begin{cases} -2x - 1, & x \leq -1 \\ -x, & -1 < x \leq 0 \\ x^2/2, & x > 0 \end{cases} \tag{5}$$

$$\partial f(x) = \begin{cases} \{-2\}, & x < -1 \\ [-2, -1], & x = -1 \\ \{-1\}, & -1 < x < 0 \\ [-1, 0], & x = 0 \\ \{x\}, & x > 0 \end{cases}$$



Fermat's Rule: $x \in \arg\min_x f(x) \Leftrightarrow 0 \in \partial f(x)$

Relating Normal Cones and Subgradients

For convex $\Omega \subset \mathbb{R}^n$ we define the *indicator function*:

$$I_\Omega(x) = \begin{cases} 0 & \text{if } x \in \Omega \\ +\infty & \text{otherwise.} \end{cases}$$

Thus the constrained optimization problem

$$\min f(x) \quad \text{s.t. } x \in \Omega$$

can be written as

$$\min f(x) + I_\Omega(x).$$

Theorem

For a convex set $\Omega \subset \mathbb{R}^n$, we have that $N_\Omega(x) = \partial I_\Omega(x)$ for all $x \in \Omega$.

Proof.

Given $v \in N_\Omega(x)$, we have

$$I_\Omega(y) - I_\Omega(x) = 0 - 0 = 0 \geq v^T(y - x), \quad \text{for all } y \in \Omega,$$

and

$$I_\Omega(y) - I_\Omega(x) = \infty - 0 = \infty \geq v^T(y - x), \quad \text{for all } y \notin \Omega.$$

Thus $v \in \partial I_\Omega(x)$. Supposing now that $v \in \partial I_\Omega(x)$, we have

$$0 = I_\Omega(y) \geq I_\Omega(x) + v^T(y - x) = v^T(y - x), \quad \text{for all } y \in \Omega,$$

which implies that $v \in N_\Omega(x)$. □

Optimality

If f is smooth convex function and Ω is a closed convex set, then x minimizes $f(x) + I_\Omega(x)$ if

$$0 \in \partial(f(x) + I_\Omega(x)) = \nabla f(x) + N_\Omega(x).$$

Thus, the following is a sufficient optimality condition for the problem $\min_{x \in \Omega} f(x)$:

$$-\nabla f(x) \in N_\Omega(x).$$

Special case of $\Omega = \mathbb{R}^n$: a sufficient condition for optimality when f is a smooth convex function is

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

If f is not convex, $\nabla f(x) = 0$ is still a necessary condition.

Optimization Taxonomy

- Linear (and Quadratic) vs Nonlinear
- Smooth vs Nonsmooth
- Continuous vs Discrete
- Convex vs Nonconvex
- Combinatorial
- Networks
- Conic
- Matrix (including semidefinite programming)
- Stochastic

References I

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, New York.