Overview of optimization concepts

Jitkomut Songsiri

Department of Electrical Engineering
Faculty of Engineering
Chulalongkorn University

CUEE

August 28, 2023

Outline

Math background

Required knowledge

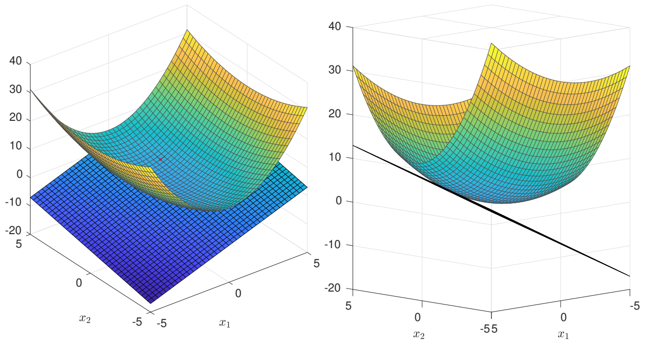
please review backgrounds on

- linear algebra with keywords:
  - system of linear equations, over-determined/under-determined, square systems
  - basic algebraic operations of vectors and matrices
  - vector and matrix norms
  - structured matrices (diagonal, symmetric, triangular, positive definite)
  - eigenvalue and eigenvector
- calculus of several variables with keywords:
  - contour, gradient, Jacobian, Hessian
  - limit, continuity, differentiability
  - sequence, convergence
- visualization of functions of several variables (surface, contour, tangent)

Tangent plane

a tangent plane of $f(x)$ at x_0 is obtained by the first-order Taylor approximation

$$f(x) \approx f(x_0) + \nabla f(x_0)^T (x - x_0)$$



$$f(x) = x_1^2 + (1/4)x_2^2$$

$$x_0 = (1, 2), \ \nabla f(x_0) = (2, 1)$$

plane: $2 + 2(x_1 - 1) + (x_2 - 2) = 0$

the gradient of f is the normal vector of the tangent plane

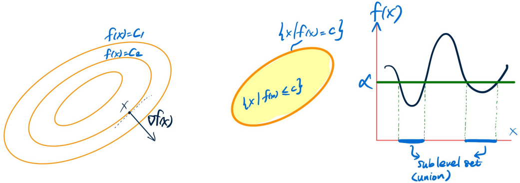
Contour and level set

definitions:

- a **contour** of a function f is $\{x \in \mathbf{R}^n \mid f(x) = \alpha\}$

  (also called a **level set** of f corresponding to α)
- a **sublevel set** of f corresponding to a value α is

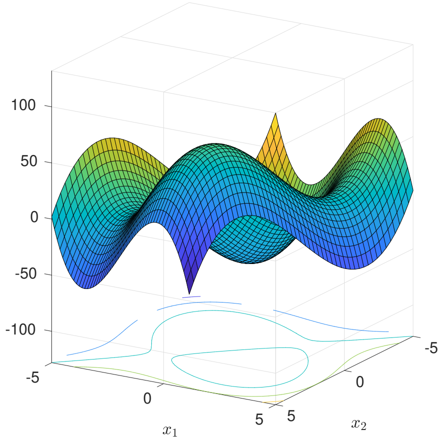
$$S_\alpha = \{x \in \mathbf{R}^n \mid f(x) \leq \alpha\}$$

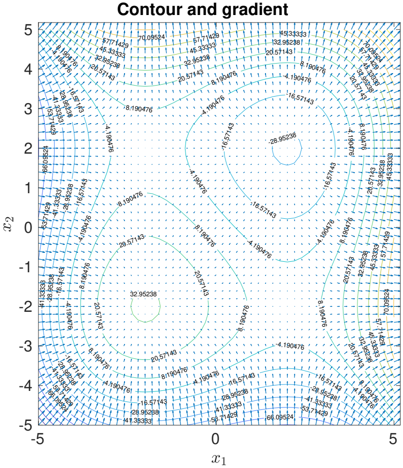


- $\nabla f(x)$ is orthogonal to the tangent line of the surface

$\nabla f(x)$ is the rate of change in f; hence, ∇f points to the direction that $f(x)$ increases

$f(x) = 2 - 12(x_1 + x_2) + x_1^3 + x_2^3$ (f has a local maximum and minimum)





notice the gradient directions toward the local maximum and minimum

System of linear equations

a system of linear equations can be represented in a matrix form

$$y = Ax$$

setting: given $y \in \mathbf{R}^m$ and $A \in \mathbf{R}^{m \times n}$, find x that satisfies the equations

- square system $(m = n)$: a solution exists and unique if A is invertible
- tall system $(m > n)$: the existence of solution depends on A, y whether $y \in \mathcal{R}(A)$
- fat system $(m < n)$: if a solution exists, then there are many solutions

  if x_p is a particular solution, and $z \in \mathcal{N}(A)$ then $x = x_p + z$ is a general solution

in optimizaiton context, linear equality constraints are usually given as a fat system

$$\{ \, x \in \mathbf{R}^n \mid \sum_{i=1}^{n} x_i = 1 \, \}$$

Linear function

a linear function $f : \mathbf{R}^n \to \mathbf{R}$ is of the form

$$f(x) = a^T x = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n$$

- $a = (a_1, a_2, \ldots, a_n)$ is a given parameter
- the contour of f is a hyperplane with the normal vector a
- $\nabla f(x) = a$ (constant, not depend on x)
- for $b \neq 0$, $f(x) = a^T x + b$ is called an affine function

the concept can be extended to a function of matrices: $f : \mathbf{R}^{m \times n} \to \mathbf{R}$

$$f(X) = \mathbf{tr}(A^T X) = \sum_{ij} a_{ij} x_{ij}$$

conceptually, f is a *linear* function of each entry in the variable

Quadratic function

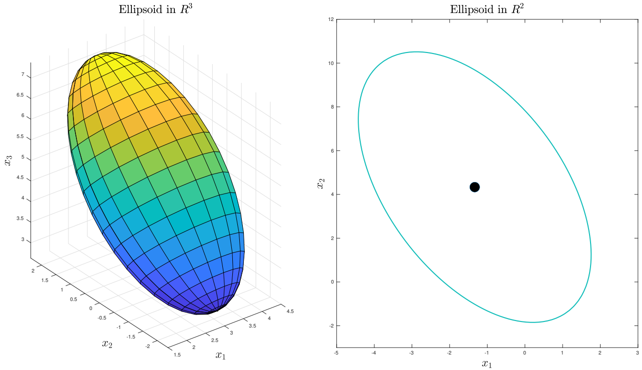
given $P \in \mathbf{R}^{n \times n}, q \in \mathbf{R}^n, r \in \mathbf{R}$, a quadratic function $f : \mathbf{R}^n \to \mathbf{R}$ is of the form

$$f(x) = (1/2)x^T P x + q^T x + r$$

- $x^T P x$ is aka an energy form (due to the quadratic form that appears in the energy/power of some physical variables)
- ✎ verify that $x^T P x = \frac{x^T (P+P^T) x}{2}$; then the energy term only takes the symmetric part of P; hence, we often consider $P \in \mathbf{S}^n$ (P is assumed to be symmetric later on)
- $\nabla f(x) = Px + q$ (derivative of quadratic function becomes linear)
- the contour shape of f depends on the property of P (pdf, indefinite, magnitude of eigenvalues, direction of eigenvectors)

Quadratic function (positive definite)

let $f(x) = (1/2)x^T P x + q^T x$ where $P \succ 0$



since P is invertible, we can complete the square

$$f(x) = (1/2)[(x + P^{-1}q)^T P (x + P^{-1}q) - q^T P^{-1} q]$$

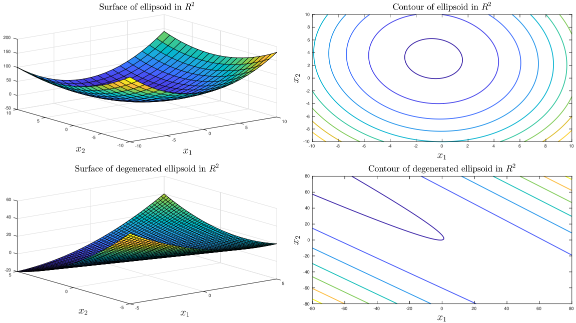
ellipsoid parametrized by P^{-1} with center at $-P^{-1}q$

Quadratic function (positive semidefinite)

let $f(x_1, x_2) = (1/2)(x^T P x) + q^T x$ with $q = (1, -3)$ and two cases of P

$$P = \begin{bmatrix} 2 & 0.1 \\ 0.1 & 1 \end{bmatrix} \succ 0$$

$$P = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \succeq 0$$

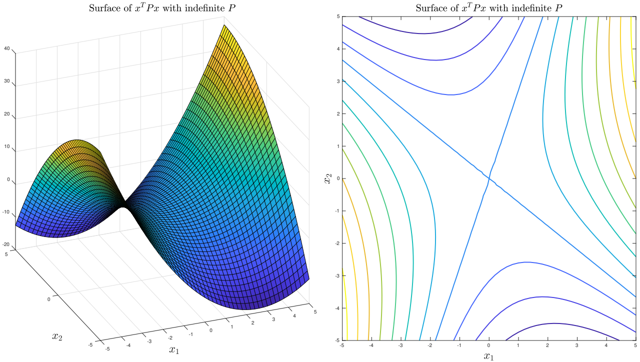


- $P \succ 0$: sublevel set of f is bounded (region inside the ellipsoid)
- $P \succeq 0$: sublevel set of f is unbounded

  (if $x = t(1, -1) \in \mathcal{N}(P)$ then $f(x) = tq^T(1, -1) = 4t \to -\infty$ by choosing $t \to -\infty$)

Quadratic function (indefinite)

let $f(x_1, x_2) = (1/2)(x^T P x) + q^T x$ with $P = \begin{bmatrix} 2 & 1 \\ 1 & -1 \end{bmatrix}$ (and invertible)



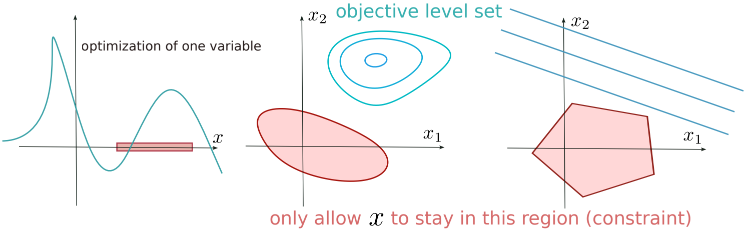
from $f(x) = (1/2)(x + P^{-1}q)^T P(x + P^{-1}q)+$ constant, we can pick t, x such that $x + P^{-1}q = tv, Pv = \lambda^- v, t \to \infty$; hence, $f(x) = t^2 \lambda^- \|v\|^2 \to -\infty$

f can be unbounded below along some direction of x

General settings

Optimization problem

an optimization is a problem of choosing a variable (x) that makes some objective function reach an extremum (can be minimum or maximum)



elements of optimization problem

- **optimization variable x:** the quantity we choose to achieve the optimization goal
- **objective function f:** a criterion that tells how objective varies upon x
- **constraints:** restrictions on x (sometimes we cannot choose x freely)

Examples of optimization

- finding a **resource allocation ratio** that maximizes the profit while the budget sum is less than a given value
- finding a **control action** to an airplane system that minimizes the deviation from the target while the control signal magnitude must be less than a value
- finding a **design** of devices/structure that minimizes the cost/weight while the size limit is from manufacturing conditions
- finding **parameters** in a model that minimizes the error between model output and observed data while the parameters must lie in a certain space, *e.g.*, all parameters are non-negative
- reconstructing a **transmitted signal** that minimizes the deviation between predicted and observed while the rate of change in the signal is bounded by a given value

Problem setting

(mathematical) optimization problem

$$
\begin{array}{ll}
\text{minimize} & f_0(x) \\
\text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\
& h_i(x) = 0, \quad i = 1, \ldots, p
\end{array}
\tag{P1}
$$

- $x = (x_1, \ldots, x_n)$: optimization variable
- $f_0 : \mathbf{R}^n \to \mathbf{R}$: objective function
- $f_i : \mathbf{R}^n \to \mathbf{R}, i = 1, \ldots, m$: inequality constraint functions
- $h_i : \mathbf{R}^n \to \mathbf{R}, i = 1, \ldots, p$: equality constraint functions

constraint set: $\mathcal{C} = \{x \in \mathbf{R}^n \mid f_i(x) \leq 0, i = 1, \ldots, m, h_i(x) = 0, i = 1, \ldots, p\}$

domain of the problem: $\mathcal{D} = \bigcap_{i=0}^{m} \mathbf{dom}\, f_i \cap \bigcap_{i=1}^{p} \mathbf{dom}\, h_i$

Optimal value

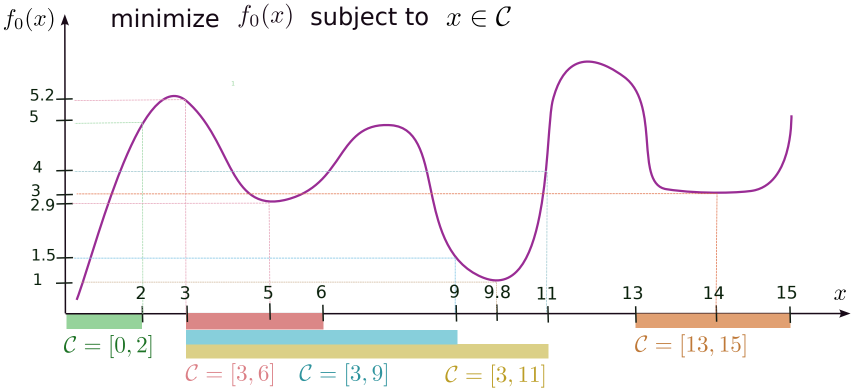
$$p^\star = \inf \{ f_0(x) \mid f_i(x) \leq 0,\ i = 1, \ldots, m,\ h_i(x) = 0,\ , i = 1, \ldots, p \}$$

- we say x is **feasible** if $x \in \mathbf{dom}\, f_0(x)$ and $x \in \mathcal{C}$
- $p^\star = \infty$ if the problem is **infeasible**
- $p^\star = -\infty$ if the problem is unbounded below
- a feasible x is called **optimal** if $f_0(x) = p^\star$; there can be many
- x is **locally optimal** if $\exists \epsilon > 0$ such that x is optimal for

$$\begin{array}{ll} \text{minimize} & f_0(z) \\ \text{subject to} & z \in \mathcal{C}, \quad \|z - x\|_2 \leq \epsilon \end{array}$$

in other words, a locally optimal point is the best solution in a neighborhood

Example



find achievable objective values, $p^\star$ and $x^\star$ for each $\mathcal{C}$

Basic examples

1. $f_0(x) = 1/x$; $p^\star = 0$, no optimal point
2. $f_0(x) = -\log x$; $p^\star = -\infty$ (unbounded below)
3. $f_0(x) = x \log x$; $p^\star = -1/e, x = 1/e$ is optimal
4. $f_0(x) = x \log x + (1-x)\log(1-x)$; $p^\star = -\log 2, x = 1/2$ is optimal
5. $f_0(x) = x^3 - 3x$; $p^\star = -\infty$, local optimum at $x = 1$
6. $f_0(x) = (x_1 - 2)^2 + (x_2 - 2)^2$; $p^\star = 0, x = (2,2)$ is optimal
7. minimize $(x_1 - 2)^2 + (x_2 - 2)^2$ s.t. $x_1 + x_2 = 2$; $\quad p^\star = 2, x = (1,1)$ is optimal
8. minimize $(x_1 - 2)^2 + (x_2 - 2)^2$ s.t. $x_1 + x_2 = 4$; $\quad p^\star = 0, x = (2,2)$ is optimal
9. minimize x_1 s.t. $x_1^2 \leq x_2, \quad x_1^2 + x_2^2 \leq 2$; $\quad p^\star = -1, x = (-1,1)$ is optimal
10. minimize $2x_1 + 2x_2$ s.t. $|x_1| + |x_2| \leq 1$; $p^\star = -2$, any x satisfying $x_1 + x_2 = -1$ is optimal (not unique)

for these examples, you can inspect a solution or find a solution in closed-form

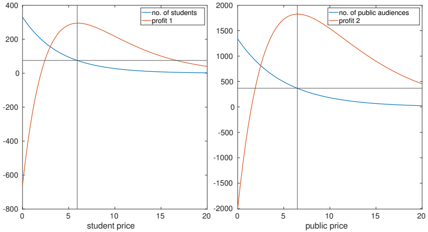
How objective and constraint functions are defined?

this is a process of *problem formulation*, motivated by an application

given: determine prices of a product for students and general audience, where the number of sold products and hence, profit vary upon the prices

setting: let $x = (x_1, x_2)$ x_1 is the price for students; x_2 is the price for general public

$$
\begin{array}{ll}
\text{maximize} & (x_1 - 2)e^{5.8-0.25x_1} + (x_2 - 1.5)e^{7.2-0.2x_2} \quad \text{(profit)} \\
\text{subject to} & e^{5.8-0.25x_1} + e^{7.2-0.2x_2} \leq 200, \quad x_1 \geq 0, \quad x_2 \geq 0
\end{array}
$$

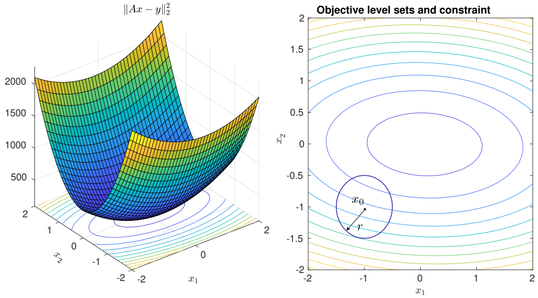


- blues: number of sold products; exponentially decrease as the price goes up
- aim to maximize the profit (as a function of prices that are non-negative)
- the objective is separable but the first constraint is not

example: given (A, y, x_0, r) as problem parameters

$$\text{minimize} \quad \|Ax - y\|_2 \quad \text{subject to} \quad \|x - x_0\|_2 \leq r$$

we aim to use a linear model Ax to approximate y while keeping such approximation valid in a norm ball



Terminology

- setting: another way of representing (P1)

$$\text{minimize} \quad f_0(x) \quad \text{subject to} \quad x \in \mathcal{C} \tag{P2}$$

- optimal point: we can also say $x^\star$ is a **global minimizer** of f_0 over $\mathcal{C}$

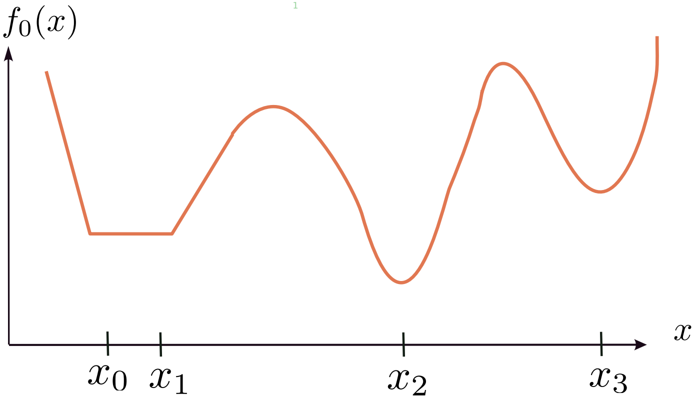
$$f_0(x) \geq f_0(x^\star) \quad \forall x \in \mathcal{C}$$

- local optimal point: we can also say $x^\star$ is a **local minimizer** of f_0 over $\mathcal{C}$

$$\exists \epsilon > 0 \quad \text{such that} \quad f_0(x) \geq f_0(x^\star) \quad \forall x \in \mathcal{C} \cap \|x - x^\star\| < \epsilon$$

  (**strict** local minimizer when $f_0(x) > f_0(x^\star)$)
- the standard form has an **implicit constraint**: $x \in \mathcal{D}$
- the constraint set $\mathcal{C}$ contains **explicit constraints**
- the problem is called **unconstrained** if it has no explicit constraints

Example



find a local/strictly local/global minimizer

Feasibility problem

a feasibility problem

$$\text{find } \; x \; \text{ subject to } \; x \in \mathcal{C}$$

can be considered as a special case of the general problem with $f_0(x) = 0$

$$\text{minimize } \; 0 \; \text{ subject to } \; x \in \mathcal{C}$$

- $p^\star = 0$ if constraints are feasible; any feasible x is optimal
- $p^\star = \infty$ if constraints are infeasible

examples: $\mathcal{C}_1$ has two-, $\mathcal{C}_2$ has infinitely many feasible points, while $\mathcal{C}_3$ is infeasible

$$\begin{aligned}
\mathcal{C}_1 &= \{x \in \mathbf{R}^2 \mid (x_1 - 1)^2 + x_2^2 = 1, x_1 + x_2 = 1\} \\
\mathcal{C}_2 &= \{x \in \mathbf{R}^2 \mid (x_1 - 1)^2 + x_2^2 \leq 1, x_1 + x_2 = 1\} \\
\mathcal{C}_3 &= \{x \in \mathbf{R}^2 \mid (x_1 - 1)^2 + x_2^2 \leq 1, x_1 + x_2 = -3\}
\end{aligned}$$

Review exercise

express the following problems in the standard form

- problem parameters: $l, u \in \mathbf{R}^n$

$$\text{minimize} \quad f_0(x) \quad \text{subject to} \quad l \preceq x \preceq u$$

- problem parameters: $A \in \mathbf{R}^{m \times n}, G \in \mathbf{R}^{p \times n}$

$$\text{maximize} \quad f_0(x) \quad \text{subject to} \quad Ax \preceq b, \; Gx = h$$

- problem parameter: $r \in \mathbf{R}^n$

$$\text{minimize} \quad \|x\|_2^2 \quad \text{subject to} \quad |x| \preceq r$$

(the notation $\preceq$ is elementwise inequality of all elements in x)

Simple conclusions about optimization

consider a constrained problem: minimize $f(x)$ subject to $x \in \mathcal{C}$ (optimal value is $p^\star$)

1. when the constraint functions are more stringent, the set $\mathcal{C}$ is smaller
2. what can you say about $p^\star$ if $\mathcal{C}$ is bigger (or smaller) ?
3. let $g(x) \leq f(x)$ for all x, and we minimize $g(x)$ subject to $x \in \mathcal{C}$; compare the new optimal value with $p^\star$
4. the problem is equivalent to **maximizing** $-f(x)$ subject to $x \in \mathcal{C}$

P1, P2, P3 are the minimization of $f(x)$ subject to $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ respectively

$$\mathcal{C}_1 = \{x \mid 0 \leq x_1, x_2 \leq 1\}, \ \ \mathcal{C}_2 = \{x \mid 1/2 \leq x_1^2 + x_2^2 \leq 1\},$$
$$\mathcal{C}_3 = \{x \mid x_1 + x_2 \leq 1, x_1 \geq 0, x_2 \geq 0\}$$

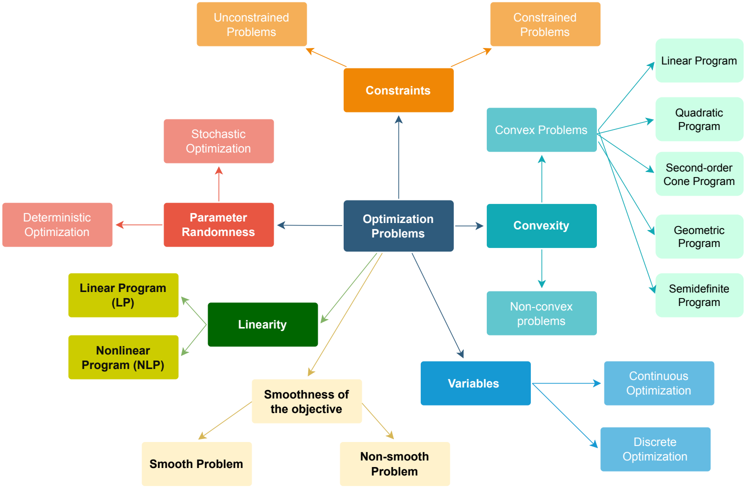
which pair of optimal values can be compared ?

Problem types

we can categorize optimization problems by

- **constraints**
  - unconstrained problem
  - constrained problems
- **variable types**
  - continuous optimization
  - discrete optimization
- **linearity of objective and constraints**
  - linear program
  - nonlinear program
- **convexity of objective and constraint set**
  - convex problem
  - non-convex problem
- **smoothness of the objective**
  - smooth problem
  - non-smooth problem
- **parameter randomness**
  - stochastic optimization
  - deterministic optimization

this course focuses on continuous and deterministic optimization



other specific problem types are integer programming, vector optimization

Unconstrained VS Constrained problems

easy example: variables in least-square problems are regarded as nonnegative values

$$\text{minimize} \quad \|Ax - b\|_2^2$$

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|_2^2 \\ \text{subject to} & x \succeq 0 \end{array}$$

- solving unconstrained problems is based on the optimality condition:

$$\nabla f_0(x) = 0$$

  find x that make the gradient zero in the cost objective (necessary condition)
- solving constrained problems depends on the type of constraint functions
  - linear equality: constraint elimination method
  - inequality equality: dedicated algorithms for some specific form

Optimality of unconstrained problems

assumption: f is twice continuously differentiable (smooth objective)

- **1st-order necessary condition:**

$$\text{if } x^\star \text{ is a local minimizer of } f \text{ then } \nabla f(x^\star) = 0$$

- **2nd-order necessary condition:** if $x^\star$ is a local minimizer of f then $\nabla f(x^\star) = 0$ and $\nabla^2 f(x^\star) \succeq 0$ (positive semidefinite)
- **2nd-order sufficient condition:** if $\nabla f(x^\star) = 0$ and $\nabla^2 f(x^\star) \succ 0$ (pdf)

$$\text{then } x^\star \text{ is a strict local minimizer of } f$$

local minimizers can be distinguished from other stationary points by examining positive definiteness of $\nabla^2 f$

example: $f(x) = x^4$ has $x^\star = 0$ as a local minimizer; $\nabla^2 f(x^\star) = 0$ (hence, 2nd-order sufficient condition fails)

Unconstrained maximization

a problem of minimizing f is equivalent to maximizing $-f$

2nd-order conditions:

- if $x^\star$ is a local maximizer of f then $\nabla f(x^\star) = 0$ and $\nabla^2 f(x^\star) \preceq 0$ (negative semidefinite)
- if $\nabla f(x^\star) = 0$ and $\nabla^2 f(x^\star) \prec 0$ (negative definite)

then $x^\star$ is a strict local maximizer of f

✎ **conclusions:**

- a point at which the gradient is zero is a stationary point (aka critical point)
- a stationary point may be a local minimizer of f, or a local maximizer, or neither, in which case it is a saddle point

Example: Rosenbrock function

given that $f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$, the gradient and Hessian of f are

$$\nabla f(x) = \begin{bmatrix} -400(x_1 x_2 - x_1^3) - 2 + 2x_1 \\ 200(x_2 - x_1^2) \end{bmatrix}, \quad \nabla^2 f(x) = \begin{bmatrix} -400(x_2 - 3x_1^2) + 2 & -400x_1 \\ -400x_1 & 200 \end{bmatrix}$$

✎ pls verify that $\nabla f(x) = 0 \Leftrightarrow x = (1, 1)$

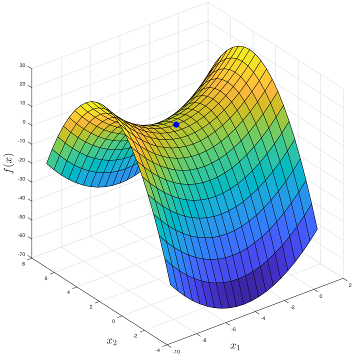
hence, $(1, 1)$ is the only stationary point and because

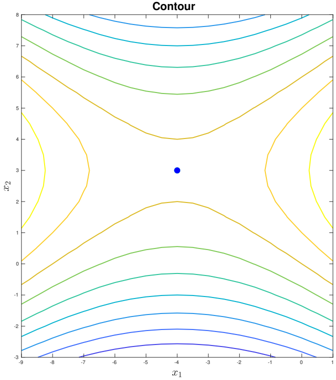
$$\nabla^2 f(1, 1) = \begin{bmatrix} 802 & -400 \\ -400 & 200 \end{bmatrix} \succ 0,$$

we conclude that $(1, 1)$ is the only local minimizer of f

Saddle point

$f(x) = 8x_1 + 12x_2 + x_1^2 - 2x_2^2$ has only one stationary point which is neither a maximum nor a minimum, but a saddle point





the stationary point is $x = (-4, 3)$

$$\nabla^2 f(x) = \begin{bmatrix} 2 & 0 \\ 0 & -4 \end{bmatrix} \not\succeq 0$$

Nonlinear least-squares (NLS)

NLS is a specific unconstrained problem of the form

$$\underset{x}{\text{minimize}} \quad f(x) := (1/2)\sum_{i=q}^{q}(r_i(x))^2$$

where $r_i : \mathbf{R}^n \to \mathbf{R}$ for $i = 1, 2, \ldots, q$

- often appear in curve fitting problems:

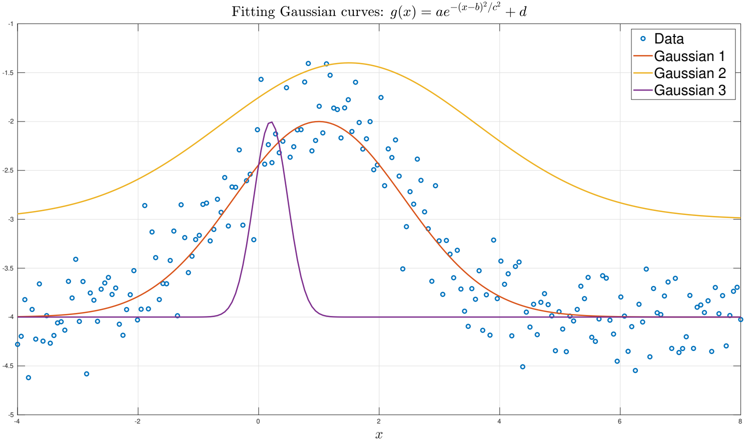
$$\text{minimize} \quad \sum_{i=1}^{N}(y_i - g(x_i))^2$$

  where g is a (nonlinear) function for fitting the data $\{(x_i, y_i)\}_{i=1}^{N}$

✎ express the minimization of $10(x_2 - x_1^2)^2 + (1 - x_1)^2$ as NLS

Nonlinear least-squares (NLS)

fitting a Gaussian curve: $g(x) = ae^{-(x-b)^2/c^2} + d$ to data points



optimization variable: $\theta = (a, b, c, d)$; explain how θ vary in the three Gaussian curves ?

Nonlinear least-squares (NLS)

gradient and Hessian of the objective function

- define $r(x) = (r_1(x), \ldots, r_m(x))$ that maps $\mathbf{R}^n \to \mathbf{R}^m$
- let $J(x) \in \mathbf{R}^{m \times n}$ be the Jacobian of r; then $\nabla f(x) = J(x)^T r(x)$
- 1st-order necessary condition is

$$\sum_{i=1}^{m} \frac{\partial r_i(x)}{\partial x} \cdot r_i(x) = 0$$

  finding a stationary point is the problem of finding roots of nonlinear equations
- by product rule, the Hessian of f is given and approximated by

$$\nabla^2 f(x) = J(x)^T J(x) + S(x) \approx J(x)^T J(x)$$

  where $S(x)$ involves the 2nd-order derivative of J

Selected problem types in applications

Selected problem types

brief concepts about the following problem types

1. convex optimization: see separate handouts (convex_optim.pdf)
2. stochastic optimization
3. nonsmooth optimization
4. scalarized multi-objective optimization
5. multi-objective optimization

What to know about convex optimization

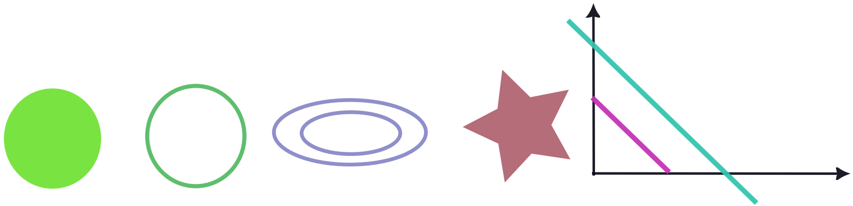
1. convex sets
2. convex functions
3. convex optimization: two common convex problems
   - linear programming
   - quadratic programming

Convex sets

a set $\mathcal{C}$ is said to be convex if for any $x, y \in \mathcal{C}$ we have

$$\theta x + (1-\theta) y \in \mathcal{C}, \quad \text{for all } \; 0 \le \theta \le 1$$

which of the following sets are convex ?



fact: an intersection of convex sets is convex (even infinitely many number of intersections)

Convex functions

convex function: $f : \mathbf{R}^n \to \mathbf{R}$ is convex if

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

for all x, y in the domain of f and $0 \leq \theta \leq 1$

loosely speaking, f is convex if it has an upward shape

examples on $\mathbf{R}$:

- affine: $ax + b$ for any $a, b \in \mathbf{R}$
- exponential: e^{ax} for any $a \in \mathbf{R}$
- powers of absolute value: $|x|^p$ for $p \geq 1$
- negative entropy: $x \log x$ on $\mathbf{R}_{++}$

Examples of convex functions on $\mathbf{R}^n$

- affine: $a^T x + b$
- norm functions: $\|x\|$
- norms of affine: $\|a^T x + b\|$
- quadratic: $x^T P x + q^T x$ when $P \succeq 0$
- negative entropy: $\sum_{i=1}^n x_i \log x_i$ on $\mathbf{R}^n_{++}$

fact: a set of inequality constraints described by convex functions is convex

$$\mathcal{C} = \{x \in \mathbf{R}^n \mid f_i(x) \leq 0,\ i = 1, 2, \ldots, m\}$$

is a convex set if all f_i's are convex functions

First- and second-order conditions of convex functions

suppose f is differentiable; then f is convex if and only if

$$\textbf{dom}\, f \text{ is convex and } f(y) \geq f(x) + \nabla f(x)^T(y-x), \quad \forall x, y \in \textbf{dom}\, f$$

- the first-order Taylor approximation of f is a **global underestimator** of f if and only if f is convex
- if $\nabla f(x) = 0$ then for all $y \in \textbf{dom}\, f$, $f(y) \geq f(x)$,*i.e.*, x is a global minimizer of f

assume that $\nabla^2 f$ exists at each point in $\textbf{dom}\, f$; then f is convex if and only if

$$\textbf{dom}\, f \text{ is convex and } \nabla^2 f(x) \succeq 0, \ \forall x \in \textbf{dom}\, f$$

f is convex if and only if its Hessian matrix is positive semidefinite

Convex programs

convex optimization problem is one of the form

$$
\begin{array}{ll}
\text{minimize} & f_0(x) \\
\text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\
& a_i^T x = b_i, \quad i = 1, \ldots, p
\end{array}
$$

where

- objective and constraint functions are **convex**
- equality constraint functions $h_i(x) = a_i^T x - b_i$ must be **affine**

result: an optimal solution of a convex program is a **global** minimizer

Linear program (LP)

a general linear program has the form

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Gx \preceq h \\ & Ax = b \end{array}$$

where $G \in \mathbf{R}^{m \times n}$ and $A \in \mathbf{R}^{p \times n}$

example: minimize the cheapest diet that satisfies the nutritional requiremenets

- $x = (x_1, \ldots, x_n)$ is nonnegative quantity of n different foods
- each food has a cost of c_j; cost objective is $c^T x$
- one unit quantity of food j contains d_{ij} amount of nutrients i
- constraints are $Dx \succeq h$ and $x \succeq 0$

Geometrical interpretation

- hyperplane: solution set of a linear equation with coefficient vector $a \neq 0$

$$\{x \mid a^T x = b\}$$

- halfspace: solution set of a linear inequality with coefficient vector $a \neq 0$

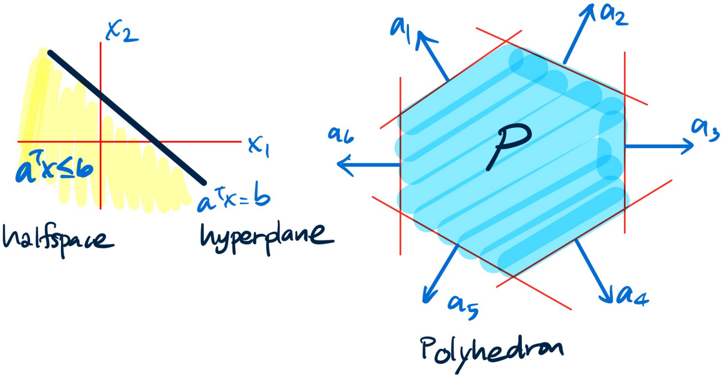
$$\{x \mid a^T x \leq b\}$$

  we say a is the **normal vector**

- polyhedron: solution set of a finite number of linear inequalities

$$\{x \mid a_1^T x \leq b_1, \;\; a_2^T x \leq b_2, \;\; \ldots, \;\; a_m^T x \leq b_m\} = \{x \mid Ax \leq b\}$$

  intersection of a finite number of halfspaces

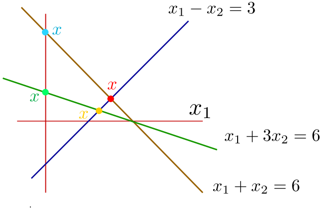


extreme point of $\mathcal{C}$

a vector $x \in \mathcal{C}$ is an extreme point (or a vertex) if we cannot find $y, z \in \mathcal{C}$ both different from x and a scalar $\alpha \in [0, 1]$ such that $x = \alpha y + (1 - \alpha) z$

Solving LPs graphically

LP 1 (left) and LP 2 (right, with non-negative constraints)



$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & x_1 + x_2 \leq 6 \\ & x_1 - x_2 \leq 3 \\ & x_1 + 3x_2 \geq 6 \end{array}$$

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & x_1 + x_2 \leq 6 \\ & x_1 - x_2 \leq 3 \\ & x_1 + 3x_2 \geq 6 \\ & x_1, x_2 \geq 0 \end{array}$$

- LP 1: feasible set is unbounded but the problem is bounded below for some c

$$c = (0,1), x^\star = \quad c = (-1,0), x^\star = \quad c = (-1,1), x^\star = \quad c = (1,3), x^\star =$$

- LP 2: feasible set is a bounded polyhedron
  - $x^\star = x$ if
  - $x^\star = x$ if
  - $x^\star$ is not unique if

$x^\star = x$ if

$x^\star = x$ if

Simple linear programs

minimize $c^T x$ over each of these simple sets

we can derive an explicit solution of these LPs

- **box constraint:** $l \preceq x \preceq u$
- **probability simplex** (or budget allocation): $\mathbf{1}^T x = 1, x \succeq 0$
- **not all budget is used:** $\mathbf{1}^T x \leq 1, x \succeq 0$
- **halfspace:** $a^T x \leq b$

draw the constraint set and inspect the solution for a given c

Some problems may not look like an LP

example 1: functions that involve ℓ_1 and ℓ_∞ norms

$$\text{minimize} \quad \|Fx - g\|_1 \quad \text{subject to} \quad \|x\|_\infty \leq 1$$

(minimize a cost measured by 1-norm having a worst-case budget constraint)

by introducing u; imposing the constraint: $-u \preceq Fx - g \preceq u$; and noting that

$$\|Fx - g\|_1 = \sum_{i=1}^{m} |f_i^T x - g_i| \leq \mathbf{1}^T u$$

the problem is equivalent to the LP

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T u \\ \text{subject to} & -u \preceq Fx - g \preceq u, \\ & -\mathbf{1} \preceq x \preceq \mathbf{1} \end{array}$$

Properties of LP

- another standard form: minimize $c^T x$ subject to $Ax = b$, $x \succeq 0$
- an LP may not have a solution (constraints are inconsistent or the feasible set is unbounded)
- we assume A is full row rank; if not, considering $Ax = b$
  - depending on A, the system could be inconsistent (hence, no extreme points), or
  - $Ax = b$ contains redundant equations, which can be removed
- if a standard LP has a finite optimal solution then

  a solution can always be chosen from among the vertices of the feasible set

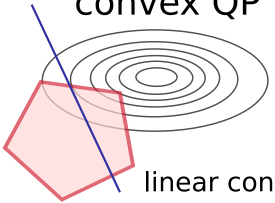
  (called **basic feasible solutions**)
- the dual of an LP is also an LP
- solutions of some simple LPs can be analytically inspected

Standard form

a **quadratic program (QP)** is in the form

$$\begin{array}{ll} \text{minimize} & (1/2)x^T P x + q^T x \\ \text{subject to} & Gx \preceq h \\ & Ax = b, \end{array}$$

where $P \in \mathbf{S}^n, G \in \mathbf{R}^{m \times n}$ and $A \in \mathbf{R}^{p \times n}$



example: constrained least-squares

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|_2^2 \\ \text{subject to} & l \preceq x \preceq u \end{array}$$

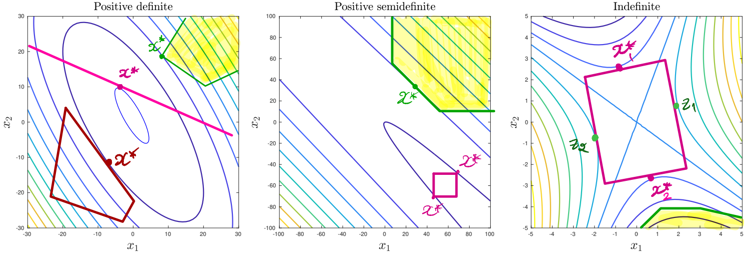
QP has **linear** constraints

Properties of QP

- an unconstrained QP is unbounded below if P is not positive definite
- an unconstrained QP has a unique solution: $x = -P^{-1}q$ when $P \succ 0$
- a QP is a convex problem if P is positive semidifinite definite
  - if $P \succeq 0$ then a local minimizer $x^\star$ is a global minimizer (by convexity)
  - if $P \succ 0$ then $x^\star$ is a *unique* global solution (by strictly convexity)
- the feasible set (polyhedron) may be empty (hence, the problem is infeasible)
- the feasible set can be unbounded (but if $P \succ 0$ it implies boundedness)
- solution of a QP may not be at a vertex
- the dual of a QP is also a QP

Contour of quadratic objective

consider three cases of P and different feasible sets



verify the location of the optimal solution for each constraint set

- left: a bounded set, a line, an unbounded feasible set
- middle: bounded and unbouded feasible sets, while f is unbounded below
- right: a bounded feasible set, while f is unbounded below and above

Applications of quadratic programming

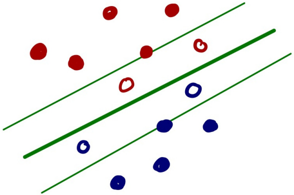
- unconstrained QP
  - least-squares
  - optimizing group representative step in k-mean clustering
- support vector machine
- control systems
- inverse problem (medical imaging, signal processing)
- least-squares with constraints (lasso and others)
- portfolio optimization

Soft-margin SVM

problem parameters: $x_i \in \mathbf{R}^n$ and $y_i \in \mathbf{R}$ for $i = 1, \ldots, N, \lambda > 0$

optimization variables: $w \in \mathbf{R}^n, b \in \mathbf{R}, z \in \mathbf{R}^N$

$$
\begin{array}{ll}
\text{minimize} & (1/2)\|w\|_2^2 + \lambda \mathbf{1}^T z \\
\text{subject to} & y_i(x_i^T w + b) \geq 1 - z_i, \quad i = 1, 2, \ldots, N \\
 & z \succeq 0
\end{array}
$$



- data are classified by separating hyperplane with maximized margin
- z_i is called a **slack variable**, allowing some of the hard constraints to be relaxed
- the problem has (convex) quadratic objective and linear constraints (QP)

Markowitz portfolio optimization

setting:

- $r = (r_1, r_2, \ldots, r_n) \in \mathbf{R}^n$; r_i is the (random) return of asset i
- the return has the mean $\bar{r}$ and covariance Σ

optimization variable: $x \in \mathbf{R}^n$ where x_i is the portion to invest in asset i

problem parameters: $\Sigma \succeq 0, \bar{r} \in \mathbf{R}^n, \gamma > 0$

$$\begin{array}{ll} \text{minimize} & -\bar{r}^T x + \gamma x^T \Sigma x \\ \text{subject to} & x \succeq 0, \quad \mathbf{1}^T x = 1 \end{array}$$

- $\mathbf{var}(r^T x) = x^T \Sigma x$ is the risk of the portfolio
- the goal is to maximize the expected return while minimize the risk
- γ is the risk-aversion parameter controlling the trade-off

Equivalent convex problems

two problems are (informally) **equivalent** if the solution of one can be obtained from the solution of the other, and vice versa

examples: P1 and P2 are equivalent (but they are not the same)

minimize	$\|Ax - y\|_2$	(P1)	minimize	$\|Ax - y\|_2^2$	(P2)
maximize	$\frac{1}{\|Ax-y\|_2}$	(P1)	minimize	$\|Ax - y\|_2^2$	(P2)
maximize	$\lvert f(x)\rvert$	(P1)	maximize	$\log \lvert f(x)\rvert$	(P2)

using monotonically increasing property of squared and log functions

Transformation that yield equivalent problems

some transformations are useful for problem re-formulation

- eliminating equality constraints
- introducing slack variables
- epigraph form
- minimizing over some variables
- using indicator function to represent constraints

Eliminating equality constraints

the problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & Ax = b \end{array}$$

is equivalent to

$$\begin{array}{ll} \text{minimize} & f_0(Fz + x_0) \\ \text{subject to} & f_i(Fz + x_0) \leq 0, \quad i = 1, \ldots, m \end{array}$$

where F and x_0 are such that

$$Ax = b \quad \Longleftrightarrow \quad x = Fz + x_0 \ \text{ for some } x_0$$

Example: eliminating equality constraints

equality constraint in the form of $Ax = b$ (non-trivial when A is fat)

$$\begin{array}{lll} \text{minimize} & \|Hx - y\|_2 & \text{(P1)} \\ \text{subject to} & x_1 + x_2 = 0 & \end{array} \qquad \begin{array}{lll} \text{minimize} & \|\tilde{H}x - y\|_2 & \text{(P2)} \\ \text{where} & \tilde{H} = \begin{bmatrix} h_1 - h_2 & h_3 & \cdots & h_n \end{bmatrix} & \end{array}$$

- find the nullspace of A and its basis vectors

$$\dim \mathcal{N}(A) = r \quad \Leftrightarrow \quad \exists F \in \mathbf{R}^{n \times r} \text{ such that } AF = 0 \text{ and } F \text{ is full column rank}$$

- find a particular solution of $Ax = b$, says x_0
- a general solutions to $Ax = b$ is expressed as $x = Fz + x_0$ for any z

Introducing slack variables

the problem

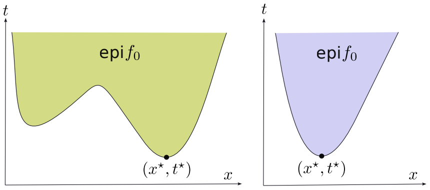
$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \end{array}$$

is equivalent to

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) + s_i = 0, \quad i = 1, \ldots, m \\ & s_i \geq 0, \quad i = 1, 2, \ldots, m \end{array}$$

Epigraph form

the epigraph of a function f_0 is the area above the graph f_0



epi $f_0 = \{(x,t); \mid x \in \mathbf{dom} f_0,\ \ f_0(x) \leq t\ \}$

the standard problem is equivalent to

$$
\begin{array}{ll}
\text{minimize (over } x, t\text{)} & t \\
\text{subject to} & f_0(x) - t \leq 0, \\
 & f_i(x) \leq 0, \quad i = 1, \ldots, m \\
 & Ax = b
\end{array}
$$

we minimize t over the epigraph of f_0 (objective is now linear of (x, t))

Example: epigraph form

example 1: $\|z\|_\infty \leq t$ if and only if $|z_i| \leq t$ for all i

$$\text{minimize}_x \quad \|Ax - y\|_\infty \quad \text{(P1)} \qquad \begin{array}{ll} \text{minimize}_{(x,t)} & t \\ \text{subject to} & -t \leq a_i^T x - y_i \leq t \quad , i = 1, \ldots, m \end{array} \quad \text{(P2)}$$

example 2: $\|Ax - y\|_1 \leq u$ if and only if $-u \preceq Ax - y \preceq u$ and $\mathbf{1}^T u \leq t$

$$\begin{array}{lll} \text{minimize}_x & \|Ax - y\|_1 & \text{(P1)} \\ \text{minimize}_{(x,u)} & \mathbf{1}^T u & \text{(P2)} \\ \text{subject to} & -u \preceq Ax - y \preceq u & \end{array}$$

Minimizing over some variables

the problem

$$\begin{array}{ll} \text{minimize} & f_0(x_1, x_2) \\ \text{subject to} & f_i(x_1) \leq 0, \quad i = 1, \ldots, m \end{array}$$

is equivalent to

$$\begin{array}{ll} \text{minimize} & \tilde{f}_0(x_1) \\ \text{subject to} & f_i(x_1) \leq 0, \quad i = 1, \ldots, m \end{array}$$

where $\tilde{f}_0(x_1) = \inf_{x_2} f_0(x_1, x_2)$

if the objective can be minimized over one variable easily, we can reduce the problem dimension

Example: minimizing over one variable

given $g_i : \mathbf{R}^n \to \mathbf{R}, y_i \in \mathbf{R}$ for $i = 1, \ldots, N$, consider the problem

$$\underset{x,d}{\text{minimize}} \quad -N \log \left[\frac{1}{d}\right] + \frac{1}{d} \sum_{i=1}^{N} (g_i(x) - y_i)^2$$

first, we can minimize over d by setting the gradient w.r.t. $1/d$ to zero

$$d = \frac{1}{N} \sum_{i=1}^{N} (g_i(x) - y_i)^2$$

the reduced problem is

$$\underset{x}{\text{minimize}} \quad \log \left[\frac{1}{N} \sum_{i=1}^{N} (g_i(x) - y_i)^2\right] \quad \Longleftrightarrow \quad \underset{x}{\text{minimize}} \quad \sum_{i=1}^{N} (g_i(x) - y_i)^2$$

Stochastic optimization

a problem is called a stochastic optimization if

- $f_i(x)$ contains some randomness, *e.g.*, problem paraters are random variables, or
- a random (Monte Carlo) choice is made in the search direction of the algorithm

example: an LP problem where c is a **random** vector

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Gx \preceq h \\ & Ax = b \end{array}$$

one way is to change the minimization objective the cost $c^T x$ is random with mean $\bar{c}^T x$ and variance

$$\mathbf{var}(c^T x) = \mathbf{var}(x^T c) = x^T \mathbf{cov}(c) x \triangleq x^T \Sigma x$$

- generally there is a trade-off between the mean and the variance
- one way is to minimize a combination of the two quantities:

$$\begin{array}{ll} \text{minimize} & \bar{c}^T x + \gamma x^T \Sigma x \\ \text{subject to} & Gx \preceq h \\ & Ax = b \end{array}$$

  where γ controls the weight between the two
- the resulting problem is an QP

Nonsmooth optimization

a function is smooth if it is differentiable and the derivatives are continuous

- example: $f(x) = |x|$ is not smooth at $x = 0$
- example: $f(x) = \|x\|$ is not smooth at $x = 0$

a problem is called **nonsmooth** if the objective or constraints are nonsmooth functions

example: lasso problems

$$\text{minimize} \quad \|Ax - b\|_2 + \gamma \|x\|_1$$

then the methods relying on the gradient should be carefully revisited

Scalarized multi-objective optimization

a common form of multi-objective problem: for a given $\gamma > 0$,

$$\text{minimize} \quad f(x) + \gamma g(x)$$

- we desire both f and g to be small but they are weighed in by a given weight, γ (or often called penalty parameter)
- as γ is higher, we penalize more on g, then the minimized g is smaller; in this case, we care less about f
- appear in model performance evaluation where two diffferent metrics are desired to be small
- example 1: minimize model error + model complexity
- example 2: minimize system tracking error + input power

Multi-objective optimization

setting: minimizing $f_0 : \mathbf{R}^n \to \mathbf{R}^m$ (vector-valued function) over a feasible set

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & x \in \mathcal{C} \end{array}$$

a vector optimization has a **vector-valued** objective function

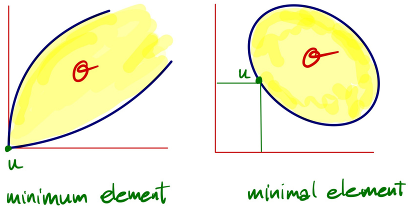
- example: $f_0(x) = $ (fuel,time) the energy used and time spent of a vehicle parameter x
- require a generalized inequality definition for comparing any two vectors of $f_0(x)$

$$\begin{bmatrix} 5 \\ 2 \end{bmatrix} \preceq \begin{bmatrix} 10 \\ 3 \end{bmatrix} \quad \text{but} \quad \begin{bmatrix} 5 \\ 2 \end{bmatrix} \not\preceq \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

here, for $f_0(x) \in \mathbf{R}^n$, we typically use the **non-negative orthant** to define $\preceq$

Achievable objective values

define $\mathcal{O} = \{ f_0(x) \mid x \in \mathcal{C} \}$ the set of objective values of feasible points



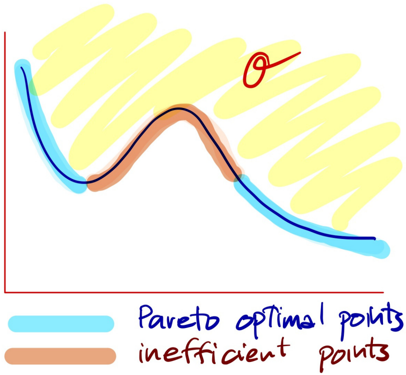
- u is said to be the **minimum** element of $\mathcal{O}$ if $u \preceq v$, for every $v \in \mathcal{O}$
- u is said to be a **minimal** element of $\mathcal{O}$ if $v \in \mathcal{O}$, $v \preceq u$ only if $v = u$
- if $\mathcal{O}$ has a minimum point (then it is unique) and

$$\exists \text{ feasible } x \quad \text{such that} \quad f_0(x) \preceq f_0(y), \quad \text{for all feasible } y$$

then we say x is **optimal**

Pareto optimal points

consider when $\mathcal{O}$ does not have a minimum element



- x is called **Pareto optimal** (or efficient) if $f_0(x)$ is a minimal element of $\mathcal{O}$
- a technique to extract pareto optimal points: scalarization (more on this later)

Optimality conditions

Unconstrained optimality

assumption: f is twice continuously differentiable (smooth objective)

- **necessary condition:** if $x^\star$ is a local minimizer of f then
  1. $\nabla f(x^\star) = 0$
  2. $\nabla^2 f(x^\star) \succeq 0$ (positive semidefinite)
- **sufficient condition:** if $\nabla f(x^\star) = 0$ and $\nabla^2 f(x^\star) \succ 0$ (positive definite), then $x^\star$ is a strict local minimizer of f
- when f is convex and differentiable, any stationary point $x^\star$ is a global minimizer of f

example: the Rosenbrock function:

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

verify that $x^\star = (1, 1)$ is the only local minimizer of f

Constrained optimality

first, define the Lagrangian function

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)$$

where λ, ν are called the **Lagrange multipliers** for inequality and equality constraints

the KKT conditions are **necessary conditions** for optimality

1. zero-gradient condition of L: $\nabla_x L(x^\star, \lambda^\star, \nu^\star) = 0$
2. primal and dual feasibility

$$f_i(x^\star) \leq 0, i = 1, \ldots, m, \quad h_i(x^\star) = 0, i = 1, \ldots, p, \quad \lambda^\star \succeq 0$$

3. complementary slackness condition: $\lambda_i f_i(x) = 0$ for $i = 1, 2, \ldots, m$

fact: for convex problems, KKT conditions are sufficient and necessary for optimality

Optimality of contrained LS

derive KKT conditions for

$$\underset{x}{\text{minimize}} \;\; (1/2)\|Ax-y\|_2^2 \;\; \text{subject to} \;\; l \preceq x \preceq u$$

the Lagrangian is $L(x,\lambda_1,\lambda_2) = (1/2)\|Ax-y\|_2^2 + \lambda_1^T(l-x) + \lambda_2^T x - u)$

KKT conditions are

1. zero-gradient of L: $A^T(Ax-y) - \lambda_1 + \lambda_2 = 0$
2. primal feasibility: $l \preceq x \preceq u$
3. dual feasibility: $\lambda_1, \lambda_2 \succeq 0$
4. complementary slackness condition:

$$\lambda_{1i}(l_i - x_i) = 0, \quad \lambda_{2i}(x_i - u_i) = 0, \quad i = 1,2,\ldots,n$$

Intro to duality theory

some quick facts

- define the dual function as the infimum of the Lagrangian over primal variables

$$g(\lambda, \nu) = \inf_{x \in \mathbf{dom}\, \mathcal{D}} L(x, \lambda, \nu)$$

- for any $\lambda \succeq 0$, the dual function provides a lower bound for $p^\star$, *i.e.*, $g(\lambda, \nu) \leq p^\star$
- any optimization problem (called a primal problem) has its dual problem

$$\text{maximize}_{\lambda, \nu} \quad g(\lambda, \nu) \quad \text{subject to} \quad \lambda \succeq 0$$

  which is the problem of finding the *best* lower bound, denoted as $d^\star$, for $p^\star$
- more theoretical results about relations between primal and dual problems – when $d^\star = p^\star$, we say we have strong duality
- solving the dual can be more beneficial in some cases

Overview of available methods

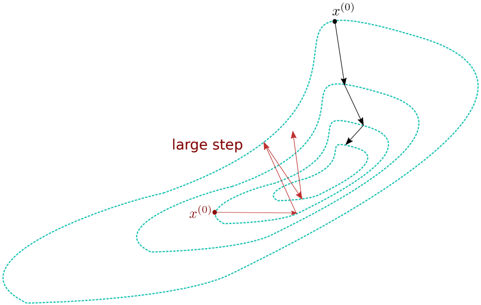
Overview of available methods

- unconstrained problems: gradient descent, Newton, quasi Newton, trust-region
- convex programs: interior point, gradient projection, ellipsoid method
- convex programs of certain structures: proximal methods
- linear programming: simplex, interior point
- quadratic programming: interior point, active set, conjugate gradient, augmented Lagrangian

Essential considerations

numerical methods are mostly iterative

- generate a sequence of points $x^{(k)}$, $k = 0, 1, 2, \dots$ that converge to a solution; $x^{(k)}$ is called the kth *iterate*; $x^{(0)}$ is the *starting point*
- computing $x^{(k+1)}$ from $x^{(k)}$ is called one *iteration* of the algorithm
- each iteration typically requires evaluations of f (or $\nabla f, \nabla f^2$) at $x^{(k)}$



- the update rule is typically of the form

$$x^{(k+1)} = x^{(k)} + t_k s^{(k)}$$

- $s^{(k)}$ is called a **search direction** and t_k is a step size

Algorithms for unconstrained problems

algorithms	search direction	meaning
steepest descent	$s^{(k)} = -\nabla f(x^{(k)})$	direction that f decreases
Newton	$s^{(k)} = -[\nabla^2 f(x^{(k)})]^{-1} \nabla f(x^{(k)})$	minimize quadratic approximation of f
quasi-Newton	$s^{(k)} = -[H^{(k)}]^{-1} \nabla f(x^{(k)})$	$H^{(k)}$ approximates the Hessian
conjugate gradient	$s^{(k)} = -\nabla f(x^{(k)}) + \beta_k s^{(k-1)}$	$s^{(k)}$ and $s^{(k-1)}$ are *conjugate* – aiming for less storage of matrices
trust-region	solution of subproblem	minimizes quadratic model with region constraint

for each iteration, the trust-region method solves for the search direction s

$$\begin{array}{ll} \text{minimize} & f(x^{(k)}) + \nabla f(x^{(k)})^T s + \frac{1}{2} s^T \nabla^2 f(x^{(k)}) s \\ \text{subject to} & \|s\| \leq \delta_k \end{array}$$

Properties of algorithms

we look at these factors when considering a method

- rate of convergence
- search direction (greatly impact the convergence)
- choice of step size (not all values is applicable)
- computational cost (storage needed, complexity)
- stopping criterion (practical conditions for checking optimality)
- descent property (objective values are monotonically decreasing)
- speed of an algorithm depends on:
  - the cost of evaluating $f(x)$ (and possibly, $\nabla f(x)$, $\nabla f^2(x)$)
  - the number of iterations required to acheive a certain accuracy

Rate of convergence

a sequence $x^{(k)}$ converges to $x^\star$ and suppose

$$\lim_{k\to\infty} \frac{\|x^{(k+1)} - x^\star\|}{\|x^{(k)} - x^\star\|} = c$$

then we obtain

convergence rate	range of c	example of $x^{(k)} \to 1$
sublinear:	$c = 1$	$x^{(k)} = 1 + \frac{1}{k+1}$
linear:	$c \in (0,1)$	$x^{(k)} = 1 + (1/2)^k$
superlinear:	$c = 0$	$x^{(k)} = 1 + (1/2)^{1.7^k}$

we say $x^{(k)}$ converges to $x^\star$ with **order** p if

$$\lim_{k\to\infty} \frac{\|x^{(k+1)} - x^\star\|}{\|x^{(k)} - x^\star\|^p} = C, \quad \text{for some } C$$

example: $x^{(k)} = 1 + (1/2)^{2^k}$ converges **quadratically** to 1

Convergence rate of algorithms

suppose $x^{(k)} \to x^\star$ (optimal solution); how fast does $x^{(k)}$ go to $x^\star$ asymptotically?

error after k iterations: typical choices are

- **Euclidean distance**: $e_k = x^{(k)} - x^\star$
- **the cost difference**: $e_k = f(x^{(k)}) - f(x^\star)$

Linear, superlinear and quadratic rate (another representation)

- **linear convergence:** there exists $c \in (0,1)$ such that

$$\|e_{k+1}\| \leq c\|e_k\| \quad \text{for sufficiently large } k$$

  - also represented as $\|e_k\| \leq Mc^k$ for $M > 0$ (converges geometrically)
  - example: $e_k = (1/2)^k$
- **superlinear convergence:** there exists a sequence c_k with $c_k \to 0$ s.t.

$$\|e_{k+1}\| \leq c_k\|e_k\| \quad \text{for sufficiently large } k$$

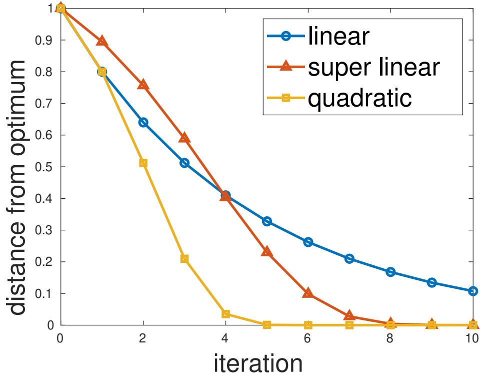
  when c_k can be further expressed as $c_k = C\beta^{p^k}$ with $C > 0, \beta \in (0,1), p > 1$, we say e_k converges superlinearly with order p (e.g., $e_k = (1/2)^{1.7^k}$)
- **quadratic convergence:** there exists a $c > 0$ s.t.

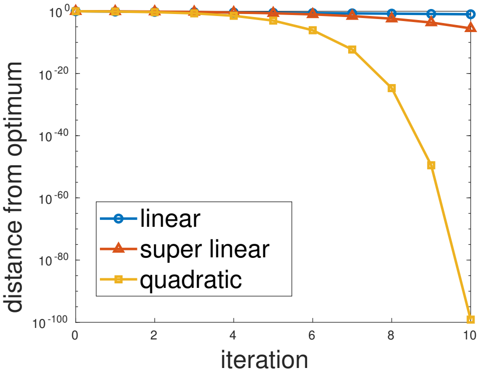
$$\|e_{k+1}\| \leq c\|e_k\|^2 \quad \text{for sufficiently large } k$$

  example: $e_k = (1/2)^{2^k}$

Examples of convergence rates

convergence rate of $(0.8)^k, C(0.8)^{1.7^k}, C(0.8)^{2^k}$ in linear and log scales





Examples of convergence analysis

what is the convergence rate of the following results (from unconstrained optimization)

$$f(x^{(l)}) - p^\star \leq \frac{2m^2}{L^2}\left(\frac{1}{2}\right)^{2^{l-n+1}} \tag{1}$$

$$f(x^{(k)}) - p^\star \leq \frac{cL\|x^{(0)} - x^\star\|^2}{k} \tag{2}$$

$$f(x^{(k)}) - p^\star \leq c^k(f(x^{(0)}) - p^\star) \tag{3}$$

$$\frac{L}{2m^2}\|\nabla f(x^{(k+1)})\|_2 \leq \left(\frac{L}{2m^2}\|\nabla f(x^{(k)})\|_2\right)^2 \tag{4}$$

(assume c, L, m are problem parameters and n is a positive integer)

- an asymptotic analysis explains what happen in the *limit* as $x^{(k)} \to x^\star$
- but, in large-scale problems, an algorithm often stops before a full convergence
- we are more interested in the accuracy of solution after k iterations presented as big $\mathcal{O}$ of some function in k

Big $\mathcal{O}$ and little o

Big $\mathcal{O}$: the notation $f(x) = \mathcal{O}(g(x))$ for $x \to c$

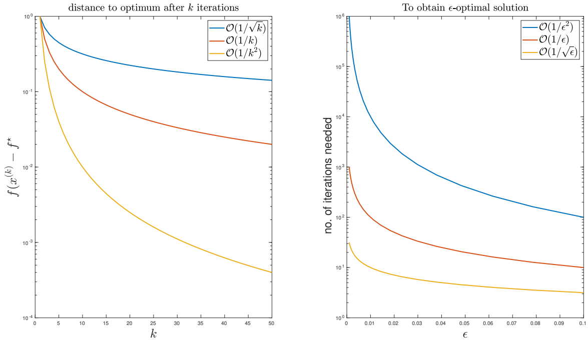
- reads "$f(x)$ has a *smaller or same* rate of growth as g when $x \to c$"
- mathematically, $\exists C > 0$ such that $|f(x)| \leq C|g(x)|$ as $x \to c$
- example: $e^x = 1 + x + \mathcal{O}(x^2)$ as $x \to 0$

little o: the notation $f(x) = o(g(x))$ for $x \to c$

- reads $f(x)$ has a *smaller* rate of growth than g when $x \to c$
- mathematically, $\lim_{x \to c} \frac{|f(x)|}{|g(x)|} = 0$
- example: $\cos x - 1 = o(x)$ as $x \to 0$

Solution precision after k iterations

there are two common ways to explain a convergence rate in large-scale problems



- the accuracy of solution after k iterations: e.g. $f(x^{(k)}) - f^\star \leq \mathcal{O}(1/k^2)$
- the number of iterations required to obtain an ϵ-optimal solution: e.g. $k \geq \mathcal{O}(\frac{1}{\sqrt{\epsilon}})$
- a constant hidden in $\mathcal{O}$ usually depends on properties of f and the distance between $x^{(0)}$ and $x^\star$

Convergence rate vs Computational cost

we prefer a fast convergence rate and less computational cost

assume n is the dimension of optimization variable and k is the number of iterations

for example, we prefer

- convergence rate: $\mathcal{O}(1/k^2) \geq \mathcal{O}(1/k) \geq \mathcal{O}(1/\sqrt{k})$
- convergence rate: $\mathcal{O}(1/\sqrt{\epsilon}) \geq \mathcal{O}(1/\epsilon) \geq \mathcal{O}(1/\epsilon)$
- cost: $\mathcal{O}(\log(n)) \geq \mathcal{O}(n) \geq \mathcal{O}(n^3)$

(by using '$X \geq Y$' we loosely mean 'prefer X to Y')

Stopping criterions

criterions rely on optimality measures

- **unconstrained optimality tolerance:** if the gradient is small enough

  absolute: $\|\nabla f(x^{(k)})\|_\infty \leq \epsilon$ **relative:** $\|\nabla f(x^{(k)})\|_\infty \leq \epsilon \|\nabla f(x^{(0)})\|_\infty$

- **constrained optimality tolerance:** $\nabla_x L$ and $\lambda_i f_i(x)$ must be small

$$\max\{ \ \|\nabla_x L(x,\lambda,\nu)\|, \ \|(\lambda_1 f_1(x), \ldots, \lambda_m f_m(x))\| \ \} \leq \epsilon$$

- **constraint tolerance:** ineq constraint should be less than zero, and equality constraint should be zero

$$f_i(x) \leq \epsilon \ \text{(close to zero)}, \ \ |h_i(x)| \leq \epsilon, \forall i$$

- **convex problem with strong duality:** if duality gap is zero

Stopping criterions

criterions based on function and step values

- **step tolerance:** difference of two consecutive steps is small

$$\text{absolute: } \|x^{(k+1)} - x^{(k)}\| \leq \epsilon \quad \text{relative: } \frac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k)}\|} \leq \epsilon$$

- **function tolerance:** the change in the objective value is small

$$\text{absolute: } |f(x^{(k+1)}) - f(x^{(k)})| \leq \epsilon \quad \text{relative: } \frac{|f(x^{(k+1)}) - f(x^{(k)})|}{|f(x^{(k)})|} \leq \epsilon$$

- **maximum number of iterations**

Optimization softwares

Numerical exercises

we will solve some small/moderate problems in class

- unconstrained problems
- nonlinear least-squares (some curve fitting problems)
- linear programs
- quadratic programs
  - trajectory control of linear system
  - least-squares with linear constraints
- constrained problems
- convex programs
  - regression problems using $\ell_2, \ell_1, \ell_\infty$-norms and huber loss
  - portfolio optimization

Exercises: Unconstrained problems

minimize the following functions

1. generate $P \succ 0, q$ randomly and let $f(x) = (1/2)x^T Px - q^T x$
2. $f(x) = \sum_{i=1}^n x_i \log x_i$
3. $f(x) = x_1^2 + x_1 x_2 + 1.5x_2^2 - 2\log(x_1) - \log(x_2)$ using initial points: $x_0 = (-1,-1), (1,1), (2,10)$
4. $f(x) = x_1^2 - x_1 x_2 + 2x_2^2 - 2x_1 + e^{x_1+x_2}$ using initial points $x_0 = (5,10), (10,10)$
5. generate $y_i \in \{1,-1\}$ and $x_i \in \mathbf{R}^n$ randomly for $i = 1,\ldots,N$ where $n = 20, N = 200$ and minimize

$$f(x) = \frac{1}{N}\sum_{i=1}^N \log\left(1 + e^{-y_i x_i^T \beta}\right) \qquad \text{soft max loss in logistic regression}$$

6. Rosenbrock function: $f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$

Exercises: Nonlinear least-squares

1. minimize $\sum_{i=1}^{N} \left(y_i - [ae^{-(x_i-b)^2/c^2} + d]\right)^2$ with variables a, b, c, d
2. minimize $\sum_{i=1}^{N} \left(y_i - \frac{K}{1+e^{-b^T x}}\right)^2$ with variables $K \in \mathbf{R}, b \in \mathbf{R}^n$

Exercises: Linear program

1. minimize $c^T x$ subject to $\mathbf{1}^T x \leq 1$, $x \succeq 0$
2. minimize $c^T x$ subject to $l \preceq x \preceq u$
3. minimize $c^T x$ subject to $\|x\|_\infty \leq 1$
4. minimize $c^T x$ subject to $0 \leq x_1 \leq x_2 \leq \cdots \leq x_n \leq 1$
5. minimize $c^T x$ subject to $d^T x = \alpha, 0 \preceq x \preceq \mathbf{1}$ with $d \succ 0$ and $0 \leq \alpha \leq \mathbf{1}^T d$
6. sparse SVM: generate $y \in \{1, -1\}$ and $x_i \in \mathbf{R}^n$ randomly for $i = 1, \ldots, N$ where $n = 20, N = 200$, set $\lambda > 0$

$$\underset{w,b}{\text{minimize}} \quad \lambda \|w\|_1 + \frac{1}{N} \sum_{i=1}^{N} \max(0, 1 - y_i(x_i^T w + b))$$

7. generate a tall $A \in \mathbf{R}^{m \times n}$ and $y \in \mathbf{R}^n$ randomly and minimize $\|Ax - y\|_1$
8. generate a tall $A \in \mathbf{R}^{m \times n}$ and $y \in \mathbf{R}^n$ randomly and minimize $\|Ax - y\|_\infty$

Exercises: Quadratic program

1. minimize $(1/2)x^T Px - q^T x$ subject to $Ax = b$ (3 cases: $P \succeq 0, P \not\succeq 0, P \preceq 0$)
2. minimize $\|Ax - y\|_2^2$ subject to (i) $\|x\|_1 \leq \alpha$ (ii) $l \preceq x \preceq u$ (iii) $x_3 = x_4 = 0$
3. soft-margin SVM: generate $y \in \{1, -1\}$ and $x_i \in \mathbf{R}^n$ randomly for $i = 1, \dots, N$

$$
\begin{array}{ll}
\text{minimize}_{w,b,z} & (1/2)\|w\|_2^2 + \lambda \mathbf{1}^T z \\
\text{subject to} & y_i(x_i^T w + b) \geq 1 - z_i, \quad i = 1, 2 \dots, N \\
& z \succeq 0
\end{array}
$$

4. given a linear system described by $y(t) = \sum_{\tau=0}^{t} h(\tau) u(t - \tau)$, $t = 0, 1, \dots, N$ where the impulse response is given as $h(t) = \frac{1}{8}(0.8)^t(1 - 0.5\cos(2t))$, design $u(0), u(1), \dots, u(N)$ to minimize

$$
\frac{1}{N+1}\sum_{t=0}^{N}(y_{\text{ref}}(t) - y(t))^2 + \frac{\lambda_1}{N+1}\sum_{t=0}^{N} u(t)^2 + \frac{\lambda_2}{N}\sum_{t=0}^{N-1}(u(t+1) - u(t))^2
$$

Exercises: Nonlinear constrained problems

1. minimize $\sum_{i=1}^{n} c_i/x_i$ subject to $a^T x = 1, x \succeq 0$ where $a, c \succ 0$
2. minimize $x_1 + x_2$ subject to $\log(x_1) + 4\log(x_2) \geq 1$
3. minimize $-2x_1 + x_2$ subject to $(1 - x_1)^3 - x_2 \geq 0$, $x_2 + 0.25x_1^2 - 1 \geq 0$ (try many choices of x_0)
4. minimize $e^{x_1 x_2 x_3 x_4 x_5} - (1/2)(x_1^3 + x_2^3 + 1)^2$ subject to

$$\sum_{i=1}^{5} x_i^2 = 10, \quad x_2 x_3 - 5x_4 x_5 = 0, \quad x_1^3 + x_2^3 + 1 = 0$$

Exercises: Convex programs

1. minimize $\|Ax - y\|_2$ subject to $\|x - x_0\| \leq \epsilon$
2. portfolio optimization:

$$\underset{x}{\text{minimize}} \quad c^T x + \gamma x^T \Sigma x \quad \text{subject to} \quad \mathbf{1}^T x = 1, \quad x \succeq 0$$

3. lasso: minimize $(1/2)\|Ax - y\|_2^2 + \gamma \|x\|_1$
4. elastic net: minimize $(1/2)\|Ax - y\|_2^2 + \gamma\{(1/2)(1-\alpha)\|x\|_2^2 + \alpha\|x\|_1\}$
5. let $p = (p_1, p_2, \ldots, p_n)$ be pmf of X where $p_k = P(X = a_k)$ for $k = 1, \ldots, n$

$$\begin{array}{ll} \text{maximize}_p & -\sum_{i=1}^n p_i \log p_i \\ \text{subject to} & -0.1 \leq \mathbf{E}[X] \leq 0.2 \\ & 0.5 \leq \mathbf{E}[X^2] \leq 0.7 \end{array}$$

use $n = 10, a = (0, 0.1, -0.2, 2, 0.5, 2, 1, -1, 0.8, -0.3)$

Unconstrained problems

MATLAB: optimization toolbox

`fminunc` uses quasi-newton and trust-region

- quasi-newton: requires description of f, uses relative optimality tolerance, relative step tolerance
- trust-region: requires description of f and ∇f, uses absolute optimality tolerance, relative function tolerance, and absolute step tolerance
- `https://www.mathworks.com/help/optim/ug/fminunc.html`

`fminsearch` uses a derivative-free method

Python: scipy.optimize

- several methods including BFGS, Newton-conjugate-gradient, trust-region Newton-conjugate-gradient, trust-region truncated generalized Lanczos, trust-region nearly exact, Nelder-Mead simplex (derivative free method)
- `https://docs.scipy.org/doc/scipy/tutorial/optimize.html`

Nonlinear least-squares

problem: minimize $r_1(x)^2 + \cdots + r_m^2(x)$ subject to $l \preceq x \preceq u$

- algorithms: trust-region reflective (default) and Levenberg-Marquardt (LM)
- for the problem without bounds, LM uses the search direction equation

$$[J(x^{(k)})^T J(x^{(k)}) + \lambda^{(k)} I] s^{(k)} = -J(x^{(k)})^T r(x^{(k)})$$

  $\lambda^{(k)}$ is called *damping parameter* (large λ, closer to gradient step)
- the nonlinear equation system $r(x) = (r_1(x), r_2(x), \ldots, r_m(x))$ is called under-determined when $m < n$

MATLAB: optimization toolbox: lsqnonlin

- trust-region reflective (default) requires that the nonlinear system $r(x) \in \mathbf{R}^q$ cannot be underdetermined, *i.e.*, $q \geq n$
- `https://www.mathworks.com/help/optim/ug/lsqnonlin.html`
- `curvefit` solves a curve fitting problem, which is an application of NLS

Python: scipy.optimize.least_squares

- trust-region reflective is suitable for large sparse problems
- LM does not handle bound constraints and it does not work for under-determined nonlinear system
- another choice: **scipy.optimize.leastsq** solves the NLS without bounds
- **scipy.optimize.curve_fit** solves a curve-fitting problem using NLS

Linear programming (LP)

MATLAB: optimization toolbox

- `linprog` uses dual-simplex and interior-point methods
- https://www.mathworks.com/help/optim/ug/linprog.html

Python: scipy.optimize.linprog

- uses interior-point and simplex methods (support sparse large-scale matrices)
- https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linprog.html

Quadratic programming

MATLAB: optimization toolbox

`quadprog` uses interior-point, trust-region reflective, and active-set methods

- interior-point only accepts convex problems
- trust-region reflective handles problems with only bounds or only linear equality constraints (not both)
- active-set handles indefinite problems only if $P \succ 0$ on $\mathcal{N}(A)$
- `https://www.mathworks.com/help/optim/ug/quadprog.html`

Python: scipy.optimize.linprog

- uses interior-point and simplex methods (support sparse large-scale matrices)
- `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linprog.html`

Constrained problems

MATLAB: optimization toolbox

`fminunc` uses several algorithms

- interior-point (default) – several ways to provide Hessian of the Lagrangian
- trust-region reflective (requires gradient)
- sequential quadratic programming (SQP) (not for large-scale)
- active-set (not for large-scale)
- `https://www.mathworks.com/help/optim/ug/fmincon.html`

Python: scipy.optimize

- several methods including trust-region and sequential least-square programming (SLSQP)
- `https://docs.scipy.org/doc/scipy/tutorial/optimize.html`

Convex problems

MATLAB: cvx

- CVX is a MATLAB-based modeling system for convex optimization
- `http://cvxr.com/cvx/`

Python

- **CVXPY**: Python-embedded modeling language for convex optimization problems available at `https://www.cvxpy.org/` by Stephen Boyd group
- **CVXOPT**: Python-based package for convex optimization available at `http://cvxopt.org/` by M. Andersen, J. Dahl and L. Vandenberghe

References

1. Chapter 1 and 2 in J. Nocedal and S.J. Wright, *Numerical Optimization*, 2nd edition, Springer, 2006
2. Chapter 1 and 2 in I. Griver, S.G. Nash, and A. Sofer, *Linear and Nonlinear Optimization*, 2nd edition, SIAM, 2009
3. S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge, 2004
4. S.Boyd, N. Parikh, E. Chu, B.Peleato, and J. Eckstein, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*, Foundations and Trends in Machine Learning, 2011