

Overview of optimization

Purdue ME 597, Distributed Energy Resources

Kevin J. Kircher

these slides draw on materials by [Stephen Boyd](#) at Stanford

Outline

Optimization problems

Optimization vocabulary

Tractable optimization problems

Our goal in studying optimization in ME 597

to become good **users** of **convex** optimization for DER applications

- optimization is a broad and deep field
- most optimization problems are intractable
- but convex problems are (usually) tractable
 - ◇ rich theory
 - ◇ efficient, reliable algorithms
 - ◇ convenient modeling software
 - ◇ often solved in subroutines for nonconvex problems
 - ◇ applications in engineering, science, economics, ...
- we won't go deep, but you can (and should!) in other classes

Optimization problems

- choose
- to minimize
- subject to
- given

Problem interpretation

- 'choose the best feasible n -vector'
- the **variable** $x = (x_1, \dots, x_n)$ is the choice made
- the **objective** $f_0(x)$ quantifies 'how bad' x is
- x is **feasible** if
 - ◇ f_0, \dots, f_m are all defined at x
(for example, $\log : \mathbf{R} \rightarrow \mathbf{R}$ is defined only for $x > 0$)
 - ◇ x satisfies all the **constraints**: $f_1(x) \leq 0, \dots, f_m(x) \leq 0$

Example: Solar photovoltaic array design

- choose solar array size (# panels or rated power) and orientation
- possible objectives:
 - ◇ initial cost (hardware, permitting, installation, ...)
 - ◇ electricity revenues or cost savings
 - ◇ pollution
- possible constraints:
 - ◇ budget
 - ◇ usable rooftop or ground area
 - ◇ panel power output equations

Example: Electric vehicle charging

- choose charging powers at each time over a planning horizon
- possible objectives:
 - ◇ electricity costs
 - ◇ pollution
 - ◇ peak electricity demand
- possible constraints:
 - ◇ battery energy and power capacities
 - ◇ battery dynamics
 - ◇ charging deadline

Equivalent problems

two problems are equivalent if

- a solution to the first readily yields a solution to the second
- and vice versa

Maximization and minimization

- suppose $g : \mathbf{R}^n \rightarrow \mathbf{R}$ quantifies 'how good' x is
- the maximization problem
 - ◇ choose $x \in \mathbf{R}^n$
 - ◇ to **maximize** $g(x)$
 - ◇ subject to $f_1(x) \leq 0, \dots, f_m(x) \leq 0$

is equivalent to the minimization problem

- ◇ choose $x \in \mathbf{R}^n$
- ◇ to **minimize**
- ◇ subject to $f_1(x) \leq 0, \dots, f_m(x) \leq 0$

Constant objective terms

for any constant $a \in \mathbf{R}$, the problem

- choose $x \in \mathbf{R}^n$
- to minimize $f_0(x) + a$
- subject to $f_1(x) \leq 0, \dots, f_m(x) \leq 0$

is equivalent to

- choose $x \in \mathbf{R}^n$
- to minimize
- subject to $f_1(x) \leq 0, \dots, f_m(x) \leq 0$

Objective and constraint transformations

- suppose
 - ◇ $h : \mathbf{R} \rightarrow \mathbf{R}$ is increasing, meaning $y > z \implies h(y) > h(z)$
 - ◇ $g_1, \dots, g_m : \mathbf{R} \rightarrow \mathbf{R}$ satisfy $g_i(y) \leq 0 \iff y \leq 0$
 - then the problem
 - ◇ choose $x \in \mathbf{R}^n$
 - ◇ to minimize $f_0(x)$
 - ◇ subject to $f_1(x) \leq 0, \dots, f_m(x) \leq 0$
- is equivalent to
- ◇ choose $x \in \mathbf{R}^n$
 - ◇ to minimize
 - ◇ subject to

Constraints with nonzero righthand sides

- for $g, h : \mathbf{R}^n \rightarrow \mathbf{R}$, the inequality constraint

$$g(x) \leq h(x)$$

is equivalent to $f_1(x) \leq 0$ with

- similarly,

$$g(x) \geq h(x)$$

is equivalent to $f_2(x) \leq 0$ with

Equality constraints

for $g, h : \mathbf{R}^n \rightarrow \mathbf{R}$, the equality constraint

$$g(x) = h(x)$$

is equivalent to the two inequality constraints

,

which are equivalent to

$$f_1(x) \leq 0 \text{ and } f_2(x) \leq 0$$

with $f_1(x) = g(x) - h(x)$ and $f_2(x) = h(x) - g(x)$

Feasibility problems

- suppose we only want to
 - ◊ find any $x \in \mathbf{R}^n$
 - ◊ satisfying $f_1(x) \leq 0, \dots, f_m(x) \leq 0$
- this is equivalent to the optimization problem
 - ◊ choose $x \in \mathbf{R}^n$
 - ◊ to
 - ◊ subject to $f_1(x) \leq 0, \dots, f_m(x) \leq 0$

Feasibility problems (example)

solving the system of nonlinear equations

$$g_1(x) = h_1(x), \dots, g_m(x) = h_m(x)$$

is equivalent to solving the feasibility problem

- find $x \in \mathbf{R}^n$
- subject to

Outline

Optimization problems

Optimization vocabulary

Tractable optimization problems

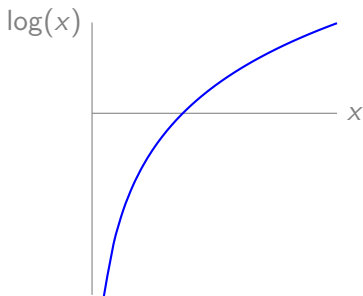
Infeasible problems

- a problem is **infeasible** if no feasible x exists
- example: minimize $x \in \mathbf{R}$ subject to

Unbounded problems

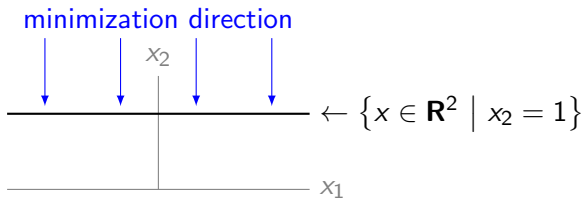
- a problem is **unbounded** if there is a sequence of feasible $x(k)$ s.t.

example: minimize $\log(x)$



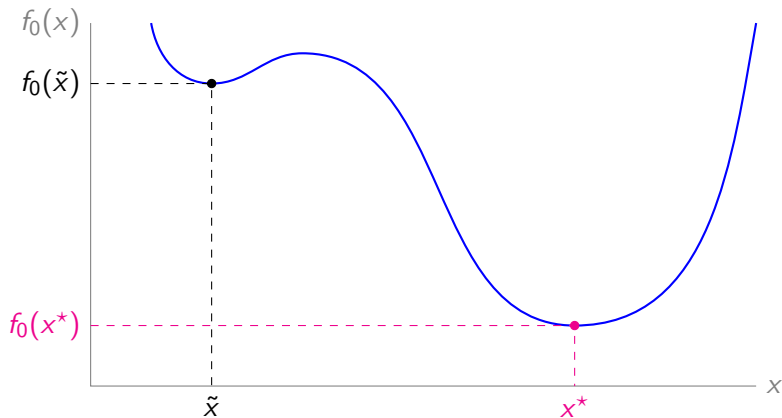
Optimality

- an $x^* \in \mathbf{R}^n$ is **optimal** (or an **optimizer**) if
 - ◊ x^* is feasible
 - ◊ $f_0(x^*) \leq f_0(x)$ for all feasible x
- infeasible problems have no optimizers
- unbounded problems have no optimizers
- feasible, bounded problems can have multiple optimizers
 - ◊ example:



Local optimality

- an \tilde{x} is **locally optimal** (or a **local optimizer**) if
 - ◊ \tilde{x} is feasible
 - ◊ $f_0(\tilde{x}) \leq f_0(x)$ for all feasible x in a neighborhood of \tilde{x}
- an unlucky local optimizer \tilde{x} might have $f_0(\tilde{x}) \gg f_0(x^*)$



Outline

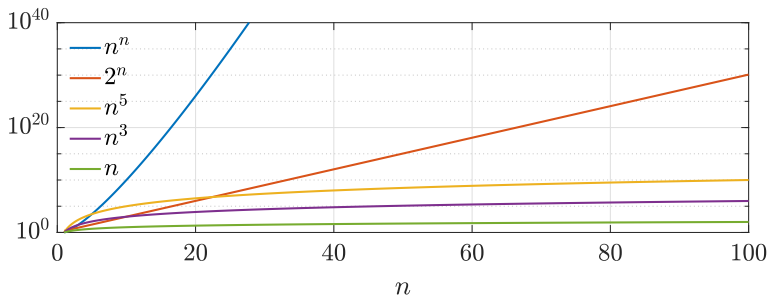
Optimization problems

Optimization vocabulary

Tractable optimization problems

Tractable optimization problems

- few optimization problems can be solved analytically
- but many can be solved numerically
- in general, *global* solve times grow exponentially in n and m
- often, *local* solve times grow only polynomially in n and m



Intractable example: The knapsack problem

- choose
- to maximize
- subject to
- given
 - ◇
 - ◇
 - ◇
- prove a polynomial-time algorithm? **earn \$1 million**

Local and global optimization

- a local optimizer \tilde{x}
 - ◇ can usually be computed efficiently
 - ◇ but might be far worse than a global x^* ($f_0(\tilde{x}) \gg f_0(x^*)$)
- a global optimizer x^*
 - ◇ gives the best feasible performance
 - ◇ but might be very slow to compute
- for **convex** problems, *all local optimizers are global optimizers*
(more on convexity next lecture)

Least squares

- choose
- to minimize
- given
- idea: no $x \in \mathbf{R}^n$ exactly satisfies all m equations in “ $Ax = b$ ”
- so least squares finds an x with $Ax \approx b$
- analytical solution:
(assuming A has linearly independent columns)
- solve time is \sim proportional to $n^2 m$

Least squares solution

- for $f(x) = \frac{1}{2} \|Ax - b\|^2$ with $P = P^\top \in \mathbf{R}^{n \times n}$,

$$\nabla f(x) =$$

- least squares has $P = A^\top A$, $q = -2A^\top b$:

$$\begin{aligned}(Ax - b)^\top (Ax - b) &= \\ &= \\ &= \end{aligned}$$

(recalling that $(CD)^\top = D^\top C^\top$ for matrices C and D)

- setting the gradient equal to zero gives

provided $A^\top A$ is invertible (**rank** $A = n$)

One least squares interpretation: Model fitting

- b_i is observation i of a **target** we want to predict (e.g., a community's electricity demand)
- A_{i1}, \dots, A_{in} are observations i of n predictive **features** (e.g., outdoor temperature, hour, weekday, season, ...)
- x_1, \dots, x_n are **parameters** in a prediction model
- problem: choose x such that
- the least squares objective

$$(Ax - b)^T (Ax - b) =$$

penalizes errors between $x_1 A_{i1} + \dots + x_n A_{in}$ and b_i for all i

Linear programming

- choose
- to minimize
- subject to
(notation: for $y, z \in \mathbf{R}^n$, $y \preceq z$ means $y_1 \leq z_1, \dots, y_n \leq z_n$)
- given $A \in \mathbf{R}^{m \times n}$, $b \in \mathbf{R}^m$, $c \in \mathbf{R}^n$
- no analytical solution, but good algorithms
- solve time is \sim proportional to $n^2 m$
- tricks can transform nonlinear problems into linear programs

Linear programming example: Chebyshev approximation

- x , A , b have same interpretations as in least squares example (parameter vector, feature matrix, target vector)
- same goal: choose x such that $x_1 A_{i1} + \dots + x_n A_{in} \approx b_i$ for all i
- instead of the least squares objective (sum of squared errors)

$$\sum_{i=1}^m (x_1 A_{i1} + \dots + x_n A_{in} - b_i)^2,$$

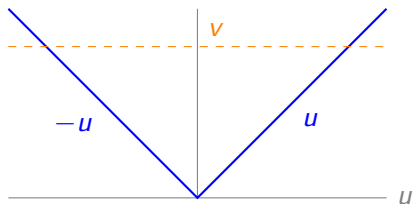
use the maximum absolute error

- this is not a linear program, but can be transformed into one

Chebyshev approximation as a linear program

- the Chebyshev approximation problem is to
 - ◊ choose $x \in \mathbf{R}^n$
 - ◊ to minimize $\max_{i=1,\dots,m} |x_1 A_{i1} + \dots + x_n A_{in} - b_i|$
- equivalently,
 - ◊ choose
 - ◊ to minimize
 - ◊ subject to
- still not a linear program, but closer

Chebyshev approximation as a linear program (continued)

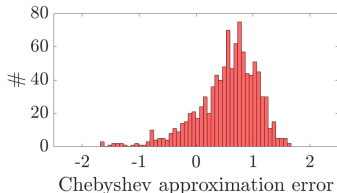
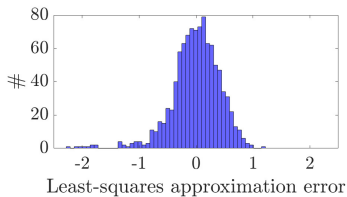
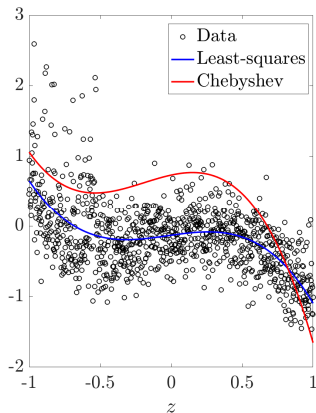


- for any $u, v \in \mathbf{R}$,
- so an equivalent problem to Chebyshev approximation is to
 - ◇ choose $(x, y) \in \mathbf{R}^{n+1}$
 - ◇ to minimize y
 - ◇ subject to

- a linear program with $n + 1$ variables and $2m$ constraints

Model fitting example

- noisy data generated from unknown function of z : $b_i = f(z_i)$
- goal: approximate each b_i by cubic, $x_1 + x_2 z_i + x_3 z_i^2 + x_4 z_i^3$
- so $n = 4$ and $A_{ij} = z_i^{j-1}$



Convex optimization

- choose
- to minimize
- subject to
- given **convex** $f_0, \dots, f_m : \mathbf{R}^n \rightarrow \mathbf{R}$
- no analytical solution, but good algorithms
- solve time is \sim proportional to $\max \{n^3, n^2 m\}$
- includes least squares, linear programming, and much more

How to use convex optimization

- formulate your problem
- hopefully, recognize it as convex
- otherwise, reformulate or approximate it as convex
- code it in a convex modeling language (CVX, CVXPY, Convex.jl, CVXR, ...)
- tell modeling language to pass your problem to a solver (SeDuMi, SDPT3, Gurobi, MOSEK, Clarabel, GLPK, ...)
- check solution, tune formulation, repeat until satisfied

Coming soon

- convex sets and functions
- solving convex optimization problems
- DER optimization examples