Overview of Convex Optimization

Jean Morrison

March 3, 2017

Resources

- Convex Optimization, Boyd and Vandenbergh:
  `web.stanford.edu/~boyd/cvxbook/`
- Software
  - CVX for Matlab: `cvxr.com/cvx/`
  - CVXPY for Python: `www.cvxpy.org/en/latest/`
  - convex.jl for Julia: `convexjl.readthedocs.io/en/latest/`
  - cvxr for R (New!): `github.com/anqif/cvxr`

- What is convexity and why does it matter?
- Properties of convex functions
- Solving convex problems

Why convexity?

Optimization problem:

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq b_i \qquad i = 1 \ldots m \end{aligned}$$

$$x \in \mathbb{R}^n$$

- optimization problems are ubiquitous (e.g. maximize the likelihood)
- in general optimization is hard
- if the problem is convex we can efficiently find a global solution!

Some familiar optimization problems

- Least squares:

$$\underset{\beta}{\text{minimize}} \qquad \|Y - A\mathbf{x}\|_2^2$$

- Lasso:

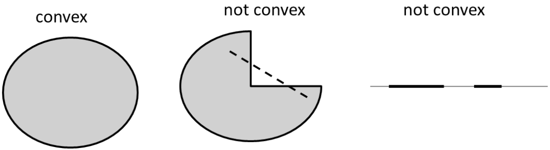
$$\underset{\beta}{\text{minimize}} \qquad \|Y - A\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1$$

- Many common probability distribution functions are log concave (i.e. negative log likelihood is convex)
  - normal
  - exponential
  - uniform
  - logistic
  - more!

Convext sets

A set $S \subseteq \mathbb{R}^n$ is convex if

$$x, y \in S, \quad \lambda, \mu \geq 0, \quad \lambda + \mu = 1$$
$$\Rightarrow \lambda x + \mu y \in S$$

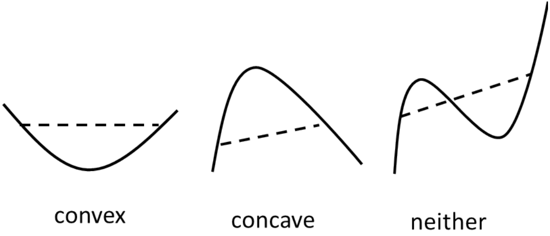


Convex functions

$f : \mathbb{R}^n \to \mathbb{R}$ is convex if **dom** f is a convex set and

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

for all $x, y \in$ **dom** f, $0 \leq \theta \leq 1$.



Examples on $\mathbb{R}$

Convex

- affine: $ax + b$ on $\mathbb{R}$ for any $a, b \in \mathbb{R}$
- exponential: e^{ax}, any $a \in \mathbb{R}$
- power: x^a on $\mathbb{R}_+$ for $a \geq 1$ or $a \leq 0$
- power of absolute value: $|x|^p$ on $\mathbb{R}$ for $p \geq 1$
- negative entropy: $x \log x$ on $\mathbb{R}_+$

Concave:

- affine: $ax + b$ on $\mathbb{R}$ for any $a, b \in \mathbb{R}$
- powers: x^a on $\mathbb{R}_+$ for $0 \leq a \leq 1$
- logarithm: $\log x$ on $\mathbb{R}_+$

Examples on $\mathbb{R}^n$ and $\mathbb{R}^{m\times n}$

- affine functions are convex and concave

$$f(x) = a^\top x + b \qquad x \in \mathbb{R}^n$$

$$f(X) = \mathbf{tr}(A^\top X) + b = \sum_{i=1}^{m} \sum_{i=1}^{n} A_{ij} X_{ij} + b \qquad X \in \mathbb{R}^{m\times n}$$

- all norms are convex:
  - $\|x\|_p = \left(\sum_{i=1}^n |x|_i^p\right)^{1/p}$ for $p \geq 1$
  - $\|x\|_\infty = \max_k |x_k|$
  - spectral (maximum singular value) norm

$$f(X) = \|X\|_2 = \sigma_{max}(X) = \left(\lambda_{max}(X^\top X)\right)^{1/2}$$

Convex problems

An optimization problem

$$\begin{array}{rll} \text{minimize} & f_0(x) & \\ \text{subject to} & f_i(x) \leq b_i & i = 1 \ldots m \end{array}$$

is convex if $f_0, f_1, \ldots, f_m$ are convex.

Ways to verify a function is convex

1. Verify the definition
2. Construct f from other convex functions using operations that preserve convexity
3. Restrict to a line and verify the definition
4. Show that $f(y) \geq f(x) + \bigtriangledown f(x)^\top (y - x)$ for all $x, y \in \textbf{dom } f$
5. Show that $\bigtriangledown^2 f(x)$ is positive semi-definite for all $x \in \textbf{dom } f$.

Ways to verify a function is convex

1. Verify the definition
Example: $f(x) = x^2$

$$[f(\theta x + (1-\theta)y)] - [\theta f(\theta) + (1-\theta)f(y)] =$$
$$\theta^2 x^2 + (1-\theta)^2 y^2 + 2\theta(1-\theta)xy - \theta x^2 - (1-\theta)y^2 =$$
$$\theta(1-\theta)(x-y)^2 \geq 0$$

Ways to verify a function is convex

2. Construct f from other convex functions using operations that preserve convexity

- composition with an affine function, non-negative weighted sum
  Example: norm of an affine function $\|Ax + b\|$
- pointwise maximum and supremum of convex functions
  Examples:
  - sum of r largest components of $x \in \mathbb{R}$

    $$f(x) = \max\{x_{i1} + x_{i2} + \cdots + x_{ir} | 1 \leq i_1 < \cdots < i_r < n\}$$

  - maximum eigenvalue of a symmetric matrix $X \in \mathbf{S}^n$:

    $$\lambda_{max}(X) = \sup_{\|y\|_2 = 1} y^\top X y$$

- composition
  $g : \mathbb{R}^n \to \mathbb{R}^k$ and $h : \mathbb{R}^k \to \mathbb{R}$

$$f(x) = h(g(x)) = h(g_1(x), \dots, g_k(x))$$

is convex if
g_i is convex, h is convex, $\tilde{h}$ nondecreasing in each argument or
g_i concave, h convex, $\tilde{h}$ nonincreasing in each argument

$$\tilde{h}(x) = \begin{cases} x & x \in \textbf{dom } h \\ \infty & \text{otherwise} \end{cases}$$

Examples:

- $exp(g(x))$ is convex if g is convex
- $\sum_{i=1}^{m} \log g_i(x)$ is concave if g_i are concave and positive

- minimization
  if $f(x, y)$ is convex in (x, y) and C is a convex set then
  $$g(x) = \inf_{y \in C} f(x, y)$$
  is convex
  Example: distance to a set $\inf_{y \in S} \|x - y\|$ is convex if S is convex
- perspective
  if $f : \mathbb{R}^n \to \mathbb{R}$ is convex then
  $$g(x, t) = tf(x/t), \qquad \textbf{dom } g = \{(x, t) | x/t \in \textbf{dom } f, t > 0\}$$
  is convex
  Example: $f(x) = x^\top x$ is convex so $g(x, t) = x^\top x/t$ is convex for $t > 0$

Ways to verify a function is convex

3. Restrict to a line and verify the definition

$f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if

$$g_{x,v}(t) = f(x + tv), \qquad \mathbf{dom}\ g = \{t | x + tv \in \mathbf{dom}\ f\}$$

is convex for all $x \in \mathbf{dom}\ f$, $v \in \mathbb{R}^n$.

Example: $f(X) = \log\det X$, $\mathbf{dom}\ X = \mathbf{S}^n_{++}$

$$\begin{aligned} g_{X,V}(t) = \log\det(X + tV) &= log \det X + \log\det\left(I + tX^{-1/2}VX^{-1/2}\right) \\ &= \log\det X + \sum_{i=1}^{n} \log(1 + t\lambda_i) \end{aligned}$$

where λ_i are eigenvalues of $X^{-1/2}VX^{-1/2}$.

g is concave in t for any $X \succ 0$, V so f is concave.

Ways to verify a function is convex

4. First order convexity condition
if f is differentiable and convex then

$$f(y) \geq f(x) + \triangledown f(x)^\top (y - x)$$

5. Second order convexity condition
if f is twice differentiable and convex then $\triangledown^2 f(x)$ is positive semidefinite for all $x \in \textbf{dom } f$.
Example: Least squares $\|Ax - b\|_2^2$

$$\triangledown f(x) = 2A^T(Ax - b) \qquad \triangledown^2 f(x) = 2A^T A \succeq 0$$

Consequences of convexity

- local optima are global
- most convex problems can be solved efficiently! (polynomial worst-case complexity)
- there are off the shelf solvers for quickly prototyping models

Modeling languages for convex optimization

CVX for Matlab, CVXPY for Python, and convex.jl (now maybe cvxr for R too!)

- use disciplined convex programming (DCP) to verify the problem is convex
- most convex problems can be written using the DCP ruleset
- use open source and commercial solves (e.g. ECOS, CVXOPT, SCS, Mosek)

DCP rules

- problems are constructed from an objective and constraints
- expressions have **curvature**, **sign**, and **monotonicity**
- you may have to get creative to write your functions so they follow the DCP rules
  - `sqrt(1 + square(x))` is not recoginzed as convex
  - `norm(vstack(1, x), 2)` is recognized as convex
- construct expressions using the library of CVXPY expressions
  `www.cvxpy.org/en/latest/tutorial/functions`

DCP rules

- objective may be
  - Minimize(convex)
  - Maximize(concave)
- constraints may be
  - affine $==$ affine
  - convex $<=$ concave
  - concave $>=$ convex

Example: regression with positive coefficients

```
from cvxpy import *
import numpy as np
n = 30
p = 10
np.random.seed(1)
truebeta = np.array([0.5, 1, 2] + [0]*7).reshape(10, 1)
X = np.random.randn(n, p)
y = np.dot(X, truebeta) + np.random.randn(n, 1)

beta = Variable(p)
objective = Minimize(sum_squares(X*beta - y))
constraints = [ beta[i] >= 0 for i in range(10)]
#or [min_entries(beta) >= 0]
prob = Problem(objective, constraints)
prob.solve()
beta.value
```

Example: mixture of normals

Suppose $y_1, \ldots, y_n$ are iid in $\mathbb{R}$ with

$$y_i \sim \pi_1 N(0, \sigma_1^2) + \pi_2 N(0, \sigma_2^2) + \pi_3 N(0, \sigma_3^2)$$

σ_1, σ_2, and σ_3 are known. We want to estimate π_1, π_2 and π_3 so that $\sum_{k=1}^3 \pi_k = 1$

$$\begin{array}{ll} \underset{\pi_1, \pi_2, \pi_3}{\text{minimize}} & \displaystyle\sum_{i=1}^n -\log\left(\sum_{k=1}^3 \pi_k \phi\left(\frac{y_i}{\sigma_k}\right)\right) \\ \text{subject to} & \displaystyle\sum_{k=1}^3 \pi_k = 1 \end{array}$$

Example: mixture of normals

$$\begin{aligned} \underset{\pi_1,\pi_2,\pi_3}{\text{minimize}} \quad & \sum_{i=1}^{n} -\log\left(A\boldsymbol{\pi}\right) \\ \text{subject to} \quad & \sum_{k=1}^{3} \pi_k = 1 \end{aligned}$$

where A is $n \times 3$ and

$$(A)_{ik} = \phi\left(\frac{y_i}{\sigma_k}\right)$$

Example: mixture of normals

Generate data

```
from cvxpy import *
import numpy as np
import scipy.stats as sts


n = 100
np.random.seed(1)
probs = [0.7, 0.2, 0.1]
sigma = [ 0.01, 0.1, 0.5]
ss = np.random.choice(sigma, size=n, p=probs)
y = np.random.normal(0, ss, n)
```

Example: mixture of normals

Set up the problem

```
#table of normal log likelihoods
A = np.empty((n, 3))
for i in range(n):
  for j in range(3):
    A[i, j] = sts.norm.pdf(y[i], 0, sigma[j])


p = Variable(3)
objective = Minimize(-1*sum(log(A*p)))
constraints = [sum(p) == 1]
prob = Problem(objective, constraints)
prob.solve()
print p.value
# 0.719 0.205 0.075
```

Algorithms

- gradient descent
- steepest descent
- newton's method
- alternating direction method of multipliers
  `http://stanford.edu/~boyd/admm.html`