Disciplined Convex Programming

Steven Diamond   Riley Murray   Philipp Schiele

SciPy 2022, July 12

Outline

Outline

Convex optimization problem — standard form

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

with variable $x \in \mathbf{R}^n$

- objective and inequality constraints $f_0, \ldots, f_m$ are convex

  for all x, y, $\theta \in [0, 1]$,

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

  i.e., graphs of f_i curve upward
- equality constraints are linear

Convex optimization problem — conic form

cone program:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b, \quad x \in \mathcal{K} \end{array}$$

with variable $x \in \mathbf{R}^n$

- linear objective, equality constraints; $\mathcal{K}$ is convex cone
- special cases:
  - linear program (LP)
  - semidefinite program (SDP)

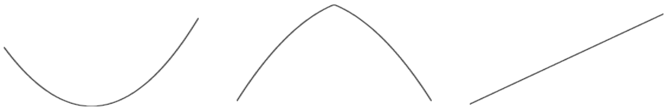
- the modern canonical form
- *there are well developed solvers for cone programs*

How do you solve a convex problem?

- use an existing custom solver for your specific problem
- develop a new solver for your problem using a currently fashionable method
  - requires work
  - but (with luck) will scale to large problems
- transform your problem into a cone program, and use a standard cone program solver
  - can be *automated* using *domain specific languages*

Outline

Curvature: Convex, concave, and affine functions



- f is *concave* if $-f$ is convex, *i.e.*, for any x, y, $\theta \in [0, 1]$,

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

- f is *affine* if it is convex and concave, *i.e.*,

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

  for any x, y, $\theta \in [0, 1]$
- f is affine $\iff$ it has form $f(x) = a^T x + b$

Examples of basic convex functions

- x^p ($p \geq 1$ or $p \leq 0$), *e.g.*, x^2, $1/x$ ($x > 0$)
- e^x
- $x \log x$
- $a^T x + b$
- $x^T P x$ ($P \succeq 0$)
- $\|x\|$ (any norm)
- $\max(x_1, \ldots, x_n)$

Examples of basic concave functions

- x^p $(0 \leq p \leq 1)$, *e.g.*, $\sqrt{x}$
- $\log x$
- $\sqrt{xy}$
- $x^T P x$ $(P \preceq 0)$
- $\min(x_1, \ldots, x_n)$

Less basic examples

Convex functions

- x^2/y for $y > 0$ and $x^T Y^{-1} x$ for $Y \succ 0$.
- $x \log(x/y)$ for $x, y \geq 0$.
- $\lambda_{\max}(X)$, for symmetric X.

Concave functions

- $\log \det X$ and $(\det X)^{1/n}$, for $X \succ 0$.
- $\log \Phi(x)$, where Φ is the Gaussian CDF.
- $\lambda_{\min}(X)$, for symmetric X.

How to verify that a function is convex or concave?

- Via the definition. For convex functions,

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

- Via first or second order conditions. For convex functions,

$$\nabla^2 f(x) \succeq 0.$$

- Via convex calculus:
  - start w/ library of basic functions that are convex or concave
  - apply transformations that preserve convexity

Convex calculus: basic rules

- **nonnegative scaling**: f convex, $\alpha \geq 0 \implies \alpha f$ convex
- **sum**: f, g convex $\implies f + g$ convex
- **affine composition**: f convex $\implies f(Ax + b)$ convex
- **pointwise maximum**: $f_1, \ldots, f_m$ convex $\implies \max_i f_i(x)$ convex
- **composition**: h convex increasing, f convex $\implies h(f(x))$ convex

. . . and similar rules for concave functions

Convex calculus: applications

- ℓ_1-regularized least-squares cost:

$$\|Ax - b\|_2^2 + \lambda\|x\|_1, \text{ with } \lambda \geq 0$$

- sum of largest k elements of x:

$$x_{[1]} + \cdots + x_{[k]}$$

- log-barrier:

$$\sum_{i=1}^{m} \log(-f_i(x)) \quad \text{on} \quad \{x \mid f_i(x) < 0\}, f_i \text{ convex.}$$

One Rule to Rule Them All

$h(f_1(x), \ldots, f_k(x))$ is convex when h is convex and for each i

- h is increasing in argument i, and f_i is convex, or
- h is decreasing in argument i, and f_i is concave, or
- f_i is affine

There's a similar rule for concave compositions
(just swap convex and concave above).

Example: The One Rule

Let's show that

$$f(u,v) = (u+1)\log\left(\frac{u+1}{\min(u,v)}\right)$$

is convex.

Three steps:

1. $x\log(x/y)$ is convex in (x,y), decreasing in y.
2. u, v are variables with $u,v > 0$.
3. $u+1$ is affine; $\min(u,v)$ is concave; both positive.

Outline

Convex Optimization

Constructive Convex Analysis

Disciplined Convex Programming (DCP)

CVXPY Tips and Tricks

Advanced Material

Extra slides

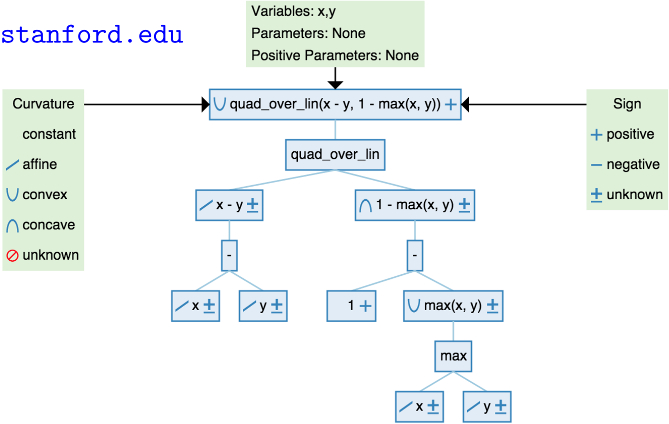
Algorithmic convexity verification: idea

- Start with an **Expression**, build a parse tree
  - Leaves: variables, constants, or parameters
  - Nodes: **Atom** objects (functions of children)
- Store curvature + monotonicity info of leaves and nodes.
  - convex, concave, affine, constant
  - increasing, decreasing
  - Helps to tag *signs*. E.g. x^2 increasing for $x \geq 0$.
- Apply The One Rule from the bottom up.

Algorithmic convexity verification: example

$$(x - y)^2/(1 - \max(x, y))$$

dcp.stanford.edu



A disciplined convex program (DCP)

- zero or one **objective**, with form
  - minimize {scalar convex expression} or
  - maximize {scalar concave expression}
- zero or more **constraints**, with form
  - {convex expression} <= {concave expression} or
  - {concave expression} >= {convex expression} or
  - {affine expression} == {affine expression}
- Convexity inferred by The One Rule and base **atoms**.

CVXPY

```
x = cp.Variable(n)
loss = cp.sum_squares(A @ x - b) + gamma*cp.norm(x,1)
prob = cp.Problem(cp.Minimize(loss),
                  [cp.norm(x,"inf") <= 1])
opt_val = prob.solve()
solution = x.value
```

- `A`, `b`, `gamma` are constants (`gamma` nonnegative)
- variables, expressions, constraints exist outside of problem
- `solve` method canonicalizes, solves, assigns `value` attributes

Exercise: limitations of CVXPY's DCP parsing

Your task:

- Find a convex CVXPY Expression `expr` for which `expr.is_dcp() == False`.

Recall: $h(f_1(x), \ldots, f_k(x))$ is convex when h is convex and for each i

- h is increasing in argument i, and f_i is convex, or
- h is decreasing in argument i, and f_i is concave, or
- f_i is affine

Exercise: limitations of CVXPY's DCP parsing

$\sqrt{x^2+1}$

- Bad: `cp.sqrt(cp.square(x) + 1)`
  - Leaves: 1 is constant, x is affine
  - Node: x^2 is convex
  - Node: $1+x^2$ is convex
  - Node: $\sqrt{\cdot}$ is concave!
- Good: `cp.norm(cp.hstack([x, 1]), 2)`.
  - $x \mapsto [x, 1]$ is affine.
  - $\|\cdot\|_2$ is convex.

Outline

A world of optimization modeling languages

- express optimization problem in high level syntax
  - declare variables
  - form constraints and objective
  - solve
- long history: AMPL, GAMS, ...
  - no special support for convex problems
  - very limited syntax
  - callable from, <u>but not embedded in</u> other languages
- DCP-based modeling: YALMIP, CVX, Convex.jl, **CVXPY**.

Using CVXPY Parameter objects

- symbolic representations of constants
- can specify sign
- change value of constant without re-parsing problem

- E.g., tuning the regularization parameter in Lasso:

```
x_values = []
for val in numpy.logspace(-4, 2, 100):
    gamma.value = val
    prob.solve()
    x_values.append(x.value)
```

Using CVXPY with Dask

```
def get_x(gamma_value):
    # return optimal x for this gamma
    return None

gammas = np.logspace(-4, 2, 30)
xs_lazy = [dask.delayed(get_x)(g) for g in gammas]
xs = dask.compute(*xs_lazy, scheduler='processes')
```

Exercise. Ridge vs. LASSO

Exercise: implement a poor man's sum-k-largest

Sum of largest k components $=$ largest sum of k components.

Approach:

- Use `itertools.combinations`
- Index into `expr` with a list of indices.
- Use `cp.sum(expr)` and `cp.maxmimum(*expr_list)`

Compare correctness to “`cp.sum_largest`“.

Outline

Convex sets

Definition: $D \subset \mathbb{R}^n$ is *convex* if

$$x, y \in D,\ \theta \in [0,1] \Rightarrow \theta x + (1-\theta)y \in D.$$

Fact: a function f is convex on D if and only if

$$\{(x,t) \in D \times \mathbb{R} \ :\ f(x) \leq t\} \text{ is convex.}$$

The DCP composition rule, revisited

Suppose $h(x, y, z)$ is decreasing in y and increasing in z.

Consider vector-valued functions G, F where

- each component of G is concave, and
- each component of F is convex.

If h is convex, then so is $h\,(x, G(x), F(x))$. *Proof:*

$$
\begin{aligned}
&\{(x, t) : h\,(x, G(x), F(x)) \le t\} \\
= &\{(x, t) : h(x, y, z) \le t,\ y \le G(x),\ F(x) \le z\}.
\end{aligned}
$$

Moral: canonicalization adds variables and constraints!

Outline

Support functions

The *support function* of a convex set D is

$$\sigma_D(x) = \max\{x^T a \,:\, a \in D\}.$$

Toy example.

The problem

$$\min\{\|x - b\|_2 \,:\, \text{if } \; \|a\|_p \le 3 \; \text{ then } \; a^T x \le 1\}$$

is equivalent to

$$\min\{\|x - b\|_2 \,:\, \sigma_D(x) \le 1\}$$

$$D = \{a : \|a\|_p \le 3\}.$$

References

- *Disciplined Convex Programming* (Grant, Boyd, Ye)
- *Graph Implementations for Nonsmooth Convex Programs* (Grant, Boyd)
- *Matrix-Free Convex Optimization Modeling* (Diamond, Boyd)

- CVX: http://cvxr.com/
- CVXPY: http://www.cvxpy.org/
- Convex.jl: http://convexjl.readthedocs.org/