Log-log convex programming

Akshay Agrawal, Steven Diamond, Stephen Boyd
Electrical Engineering, Stanford

Geometric programming

A geometric program (GP) [DPZ67] is an optimization problem

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

- $g_i : \mathbf{R}^n_{++} \to \mathbf{R}$ are monomials: $(x_1, \ldots, x_n) \mapsto c x_1^{a_1} \cdots x_n^{a_n}$, $c > 0$.
- $f_i : \mathbf{R}^n_{++} \to \mathbf{R}$ are posynomials: sums of monomials

Applications

- chemical engineering
- circuit design
- transformer design
- aircraft design
- mechanical engineering
- communications

Log-log transformation

For $f : D \to \mathbf{R}_{++}$, $D \subseteq \mathbf{R}^n_{++}$, its *log-log transformation* is $F(u) = \log f(e^u)$

Example.

For $f(x) = \sum_{i=1}^n c_i x_1^{a_{1i}} x_2^{a_{2i}} \ldots x_n^{a_{ni}}$, $c > 0$, $F(u) = \log \sum_{i=1}^n c_i \exp(a_i^T u)$

- *i.e.*, the log-log transformation of a posynomial is (the log of) a signomial
- F(u) is convex, because $c > 0$ and log-sum-exp is convex

Log-log curvature

Curvature of $F(u) = \log f(e^u)$ gives *log-log* curvature of f

- if F convex, then f is log-log convex
- if F concave, then f is log-log concave
- if F affine, then f is log-log affine

Log-log convex programs

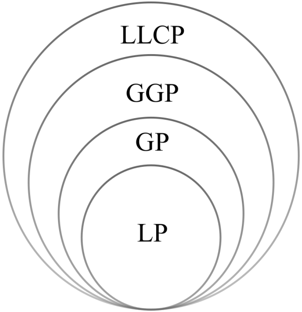
A log-log convex program (LLCP) is an optimization problem of the form

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

- $g_i : \mathbf{R}^n_{++} \to \mathbf{R}$ are log-log affine
- $f_i : \mathbf{R}^n_{++} \to \mathbf{R}$ are log-log convex

Log-log convex programs

LLCPs generalize GPs and "generalized geometric programs"



(Exponential cone programming also generalizes GP [CS17; MCW18])

Properties of log-log convex functions

Jensen's inequality

$f : \mathbf{R}^n_{++} \to \mathbf{R}_{++}$ is log-log convex iff it is convex w.r.t. the geometric mean, *i.e.*

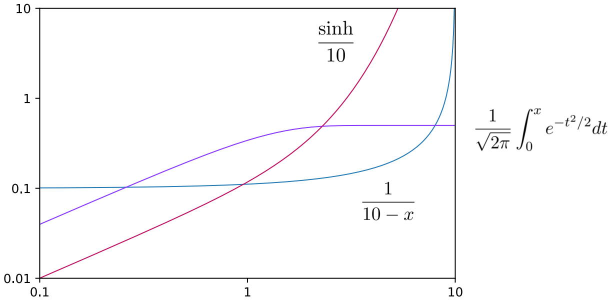
$$f(x^\theta \circ y^{1-\theta}) \leq f(x)^\theta f(y)^{1-\theta},$$

for $\theta \in [0,1]$ and $x, y \in \mathbf{dom}\, f$

- $\circ$ is the elementwise product; powers meant elementwise
- Also called geometric or multiplicative convexity
- Literature is several decades old [Mon28]
- Gives rise to interesting inequalities [Nic00]

Scalar log-log convex functions

Scalar log-log convex functions are convex on a log-log plot



Epigraph

If f is a log-log convex function, then

$$\log \mathbf{epi} f = \{(\log x, \log t) | f(x) \leq t\}$$

is a convex set. The converse is also true.

Integration

If $f : [0, a) \to [0, \infty)$ is continuous and log-log convex (log-log concave) on $(0, a)$, then

$$x \mapsto \int_0^x f(t)dt$$

is log-log convex (log-log concave) on $(0, a)$ [Mon28].

- X a random variable with continuous log-log concave density on $[0, a)$, then $P(0 < X \leq x)$ is a log-log concave function of x.
- Gaussian, Gibrat, Student's t, ... [Bar10]

Composition rule

Let

$$f(x) = h(g_1(x), g_2(x), \ldots, g_k(x)),$$

where $h : D \subseteq \mathbf{R}^k_{++} \to \mathbf{R}$ is log-log convex, and $g_i : D_i \subseteq \mathbf{R}^n_{++} \to \mathbf{R}$. Suppose for each i, one of the following holds:

- h is nondecreasing in the ith argument, and g_i is log-log convex
- h is nonincreasing in the ith argument, and g_i is log-log concave
- g_i is log-log affine

Then f is log-log convex.

Composition rule

Proof is simple: $F(u) = \log f(e^u)$ can be written as

$$H(G_1(u), G_2(u), \ldots, G_k(u)),$$

where $H(u) = \log h(e^u)$ and $G_i(i) = \log g_i(e^u)$. Result follows from the analogous composition rule for convex functions

Composition rule

Composition rule is the basis of *Disciplined Geometric Programming* (DGP), a grammar for a DSL for LLCPs [ADB18]

exactly analogous to *Disciplined Convex Programming* (DCP) [GBY06]

Examples

Log-log affine functions

Products, ratios, and powers are log-log affine

Log-log convex functions

- $x_1 + x_2$
- $\max(x_1, x_2)$
- posynomials
- ℓ_p norms
- Functions with positive Taylor expansions
- The Gamma function $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$ restricted to $[1, \infty)$

Log-log concave functions

- $x_1 - x_2$, with $x_1 > x_2 > 0$
- $-x \log x$, $x \in (0, 1)$
- complementary CDF of a log-concave density, *e.g.* $\frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt$

Functions of positive matrices

Suppose $f : \mathbf{R}_{++}^{m\times n} \to \mathbf{R}_{++}^{p\times q}$, $\mathbf{R}_{++}^{m\times n}$ denoting m-by-n matrices with positive entries
f is log-log convex if $F(U) = \log f(e^U)$ is convex w.r.t the nonnegative orthant

(log, exp meant elementwise)

Examples

- Spectral radius: $\rho(X) = \sup\{|\lambda| \mid Xv = \lambda v\}$.
- Resolvent: $f(X, s) = (sI - X)^{-1}$, $s > 0$ such that s not an eigenvalue of X.

Disciplined Geometric Programming

Disciplined geometric programming

- Analogue of DCP, but for LLCPs
- Library of atoms with known log-log curvature (sum, product, ratio, exp, ...)
- Atoms may be combined using the composition rule
- Can express LLCPs of the form

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq \tilde{f}_i(x), \quad i = 1, \ldots, m \\ & g_i(x) = \tilde{g}_i(x), \quad i = 1, \ldots, p, \end{array} \tag{1}$$

with f_i log-log convex, $\tilde{f}_i$ log-log concave, g_i and $\tilde{g}_i$ log-log affine (must be verifiable by the composition rule)

Implementation

- DGP implemented as a reduction in CVXPY 1.0 [AVD+18]: `https://www.cvxpy.org/tutorial/dgp/index.html`
- User types in a DGP-compliant LLCP and calls a single method to solve it
- CVXPY reduces the LLCP to a (disciplined) convex program, solves it, and returns a solution to the original problem

Example

```
import cvxpy as cp

X = cp.Variable((3, 3), pos=True)
objective_fn = cp.pf_eigenvalue(X)
known_value_indices = tuple(zip(*
  [[0, 0], [0, 2], [1, 1], [2, 0], [2, 1]]))
constraints = [
  X[known_value_indices] == [1.0, 1.9, 0.8, 3.2, 5.9],
  X[0, 1] * X[1, 0] * X[1, 2] * X[2, 2] == 1.0,
]
problem = cp.Problem(cp.Minimize(objective_fn), constraints)
problem.solve(gp=True)
print("Optimal value: ", problem.value, " Solution: ", X.value)
```

References

[ADB18] A. Agrawal, S. Diamond, and S. Boyd. Disciplined Geometric Programming. *arXiv* (2018). arXiv: 1812.04074 [math.OC].

[AVD+18] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision* 5.1 (2018), pp. 42–60.

[Bar10] Á. Baricz. Geometrically concave univariate distributions. *Journal of Mathematical Analysis and Applications* 363.1 (2010), pp. 182 –196.

[CS17] V. Chandrasekaran and P. Shah. Relative entropy optimization and its applications. *Mathematical Programming* 161.1-2 (2017), pp. 1–32.

[DPZ67] R. Duffin, E. Peterson, and C. Zener. *Geometric Programming — Theory and Application*. New York, 1967.

[GBY06] M. Grant, S. Boyd, and Y. Ye. Disciplined convex programming. In *Global optimization*. Springer, 2006, pp. 155–210.

[Mon28] P. Montel. Sur les fonctions convexes et les fonctions sousharmoniques. *Journal de Mathématiques Pures et Appliquées* 9.7 (1928), pp. 29–60.

[MCW18] R. Murray, V. Chandrasekaran, and A. Wierman. Newton polytopes and relative entropy optimization. *arXiv* (2018). arXiv: 1810.01614 [math.OC].

[Nic00] C. Niculescu. Convexity according to the geometric mean. *Mathematical Inequalities and Applications* 3.2 (2000), pp. 155–167.