Constructive Convex Analysis and Disciplined Convex Programming

Mathématiques Informatique et Statistique Appliquées (MISA)
Université d'Antananarivo

Constructive convexity verification

- Start with a function given as expression
- Build a parse tree for the expressionl
  - leaves are variables or constants
  - nodes are functions of children, following the general rule
- Tag each subexpression's curvature (convex, concave, affine, constant) and sign
- This is sufficient but not necessary for convexity.

Disciplined convex programing (DCP)

- A framework for describing convex optimization problems
- It is based on constructive convex analysis
- It is sufficient but not necessary for convexity
- A basis for several domain specific languages and tools for convex optimization

Structure of a DCP

- zero or one objective:
  - minimize {scalar convex expression}or
  - maximize {scalar concave expression}
- zero or more constraints:
  - {convex expression} <= {concave expression} or
  - {concave expression} >= {convex expression} or
  - {affine expression} == {affine expression}

Expressions

- Expressions are formed from
  - variables
  - constants
  - functions
- Library functions have known convexity, monotonicity, and sign properties
- All subexpressions match general composition rule

Modeling languages for convex optimization

These are all based on DCP :

- CVX (Matlab)
- CVXPY (Python)
- Convex.jl (Julia)
- CVXR (R)