Regarding CVX

The modeling language CVX is designed for certain kinds of convex optimization problems, that is, certain problems of the form

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in S \ , \end{array}$$

where S is a convex set[1] and f is a convex function[2]. Also said to be convex optimization problems are problems of the form

$$\begin{array}{ll} \max & g(x) \\ \text{s.t.} & x \in S \ , \end{array}$$

where again S is a convex set, but where g is a concave function[3]. CVX consists of various building blocks for creating sets S, and functions f and g, which result in convex optimization problems that can be handled by available solvers.

The CVX User's Guide lists built-in functions in sections 9.2 and 9.3 (and lists built-in sets in section 9.4). Note that at the end of each function's description is stated whether the function is convex or concave. For example, `sum_largest(x,5)` is a convex function and thus can be the objective function in a minimization problem (whereas $x \mapsto -\texttt{sum\_largest}(\texttt{x},5)$ is concave and can be the objective in a maximization problem).

The only functions that are both convex and concave are the affine functions, that is, functions of the form $x \mapsto c^T x + d$, where c is a vector and d is a number. (In case c is the zero vector, the function has the same output d for all inputs x, i.e., a constant function.)

If two convex (concave) functions are added together, the resulting function is convex (concave). If a convex (concave) function is multiplied by a positive number, the resulting function is convex (concave). Thus, for example, the function $x \mapsto \texttt{sum\_largest}(\texttt{x},5) + \texttt{10} * \texttt{norm}(\texttt{x})$ could be the objective function in a CVX minimization problem.

An especially useful fact is that composing a convex function with an affine transformation gives a convex function, that is, if f is convex, if A is a matrix, and b is a vector, then $x \mapsto f(Ax+b)$ is a convex function. Thus, for example, $\texttt{x} \mapsto \texttt{sum\_largest}(\texttt{A} * \texttt{x} + \texttt{b}, 5)$ is a convex function (whose output is the sum of the five largest quantities $\alpha_i^T x + b_i$, where α_i^T is the i^{th} row of A).

Rather than relying explicitly on the convex sets listed in section 9.4, most of your modeling will instead build convex sets by making use of convex and concave functions. The critical observation is that if f is a convex function and g is a concave function, then the set $\{x : f(x) \leq g(x)\}$ is convex. Thus, for example, an appropriate constraint in CVX is

$$\texttt{sum\_largest}(\texttt{x},5) <= \texttt{log\_prod}(\texttt{x}) \ ,$$

as is

$$\texttt{2} <= \texttt{log\_prod}(\texttt{x}) \quad \text{and} \quad \texttt{sum\_largest}(\texttt{x},5) <= \texttt{9}$$

(remember that affine functions – and hence the constant functions – are both convex and concave, and thus can go on either side of inequalities).

A property you will use extensively is that the intersection of convex sets is convex. Thus, whereas each of the three inequalities above defines its own convex set, the set consisting of all x which simultaneously

[1] A set $S \subseteq \mathbb{R}^n$ is said to be "convex" if for every pair of points $x, y \in S$, the line segment connecting x to y is entirely contained in S, that is,

$$\text{whenever } x, y \in S \text{ and } 0 \leq t \leq 1 \text{ then } tx + (1-t)y \in S \ .$$

[2] A function f is said to be "convex" if its epigraph is a convex set, where

$$\text{epigraph } = \{(x,t) : t \geq f(x)\}$$

(i.e., the set consisting of the graph of f and everything above it).

[3] A function g is said to be "concave" if the function obtained by multiplying output by -1 is a convex function, that is, if the function $x \mapsto -g(x)$ is a convex function.

satisfy the inequalities also is a convex set. To require x to be in this set, simply include all three of the inequalities in the body of a CVX model.

For equation constraints (as opposed to inequality constraints), only affine functions can appear. Thus `sum(x) == 8` is fine, but `sum_largest(x,5) == 8` is not.

Constraints do not necessarily have to be written individually. For example, if an $m \times n$ matrix `A` has been specified, and also a right-hand side vector `b` $\in \mathbb{R}^m$, then writing `A*x <= b` enforces all m of the individual inequalities.