SCS - Splitting Conic Solver

An R interface to the Splitting Conic Solver (SCS).

Definitions

SCS solves the following problem:

primal		*dual*
$$\begin{array}{lrl} \min_x & c^\top x & \\ s.t. & Ax + s & = b \\ & (x, s) & \in \mathbb{R}^n \times \mathcal{K} \end{array}$$	Equation (1)	$$\begin{array}{lrl} \max_y & -b^\top y & \\ s.t. & -A^\top y + r & = c \\ & (r, y) & \in \{0\}^n \times \mathcal{K}^* \end{array}$$ (1)

Where the cone $\mathcal{K}$ can be any Cartesian product of the following cones:

Name	Definition
zero cone	$\{x \vert x = 0\}$ (dual to the free cone$\{x \vert x \in \mathbb{R}\}$)
positive orthant	$\{x \vert x \geq 0\}$
second-order cone	$\{(t, x) \mid \vert\vert x\vert\vert_2 \leq t, x \in \mathbb{R}^n, t \in R\}$
positive semidefinite cone	$\left\{X \mid min(eig(X)) \geq 0,\ X = X^T,\ X \in \mathbb{R}^{n \times n}\right\}$
exponential cone	$\left\{(x, y, z) \mid ye^{\frac{x}{y}} \leq z,\ y > 0\right\}$
dual exponential cone	$\left\{(u, v, w) \mid -ue^{\frac{v}{u}} \leq ew, u < 0\right\}$
power cone	$\left\{(x, y, z) \mid x^a * y^{(1-a)} \geq \vert z\vert,\ x \geq 0,\ y \geq 0\right\}$
dual power cone	$\left\{(u, v, w) \mid \left(\frac{u}{a}\right)^a * \left(\frac{v}{(1-a)}\right)^{(1-a)} \geq \vert w\vert,\ u \geq 0,\ v \geq 0\right\}$

Usage

```r
scs(A, b, obj, cone, control)
```

Important Note

The order of the rows in matrix A has to correspond to the order given in the table "Cone Arguments", which means means rows corresponding to *primal zero cones* should be first, rows corresponding to *non-negative cones* second, rows corresponding to *second-order cone* third, rows corresponding to *positive semidefinite cones* fourth, rows corresponding to *exponential cones* fifth and rows corresponding to *power cones* at last.

Arguments

A	a matrix of constraint coefficients
b	a numeric vector giving the primal constraints
obj	a numeric vector giving the primal objective
cone	a list giving the cone sizes
control	a list giving the control parameters

Cone Arguments

Symbol	Type	Length	Description
`f`	integer	1	number of primal zero cones (dual free cones), which corresponds to the primal equality constraints
`l`	integer	1	number of linear cones (non-negative cone)
`q`	integer	≥ 1	vector of second-order cone sizes
`s`	integer	≥ 1	vector of positive semidefinite cone sizes
`ep`	integer	1	number of primal exponential cones
`ed`	integer	1	number of dual exponential cones
`p`	numeric	≥ 1	vector of primal/dual power cone parameters

Control Arguments

Parameter	Type	Description	Default Value
`max_iters`	integer	giving the maximum number of iterations	`2500`
`normalize`	boolean	heuristic data rescaling	`TRUE`
`verbose`	boolean	write out progress	`FALSE`
`cg_rate`	numeric	for indirect, tolerance goes down like $\frac{1}{iter}^{cg\_rate}$	`2`
`scale`	numeric	if normalized, rescales by this factor	`5`
`rho_x`	numeric	x equality constraint scaling	`1e-3`
`alpha`	numeric	relaxation parameter	`1.5`
`eps`	numeric	convergence tolerance	`1e-3`

Note on Semidefinite Cones

To transform an SDP problem into the form shown in Equation (1), a half-vectorization should be performed on the matrices F_i and the strictly lower triangular values have to be scaled by $\sqrt{2}$. Furthermore to get the matrix solution an inverse transformation has to be performed on the results.

$$\begin{array}{lrl} \min_x & c^\top x & \\ s.t. & \sum_{i=1}^m x_i F_i & \succeq F_0 \\ & Ax & = b \end{array} \tag{2}$$

where $F_i \in R^{n \times n}$ are symmetric matrices, for more information see e.g. ("Vandenberghe and Boyd (1996) Semidefinite Programming" or "Andersen et al. (2011) Interior-Point Methods for Large-Scale Cone Programming")

$$F_i = \begin{pmatrix} f_{11} & f_{12} & \cdots & f_{1m} \\ f_{21} & f_{22} & \cdots & f_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ f_{m1} & \cdots & \cdots & f_{mm} \end{pmatrix}$$

$$vec(F_i) = (f_{11}, \sqrt{2}f_{21}, \ldots, \sqrt{2}f_{m1},\ f_{22}, \sqrt{2}f_{32}, \ldots, \sqrt{2}f_{m2},\ f_{m-1,m-1}, \sqrt{2}f_{m,m-1},\ f_{mm})^\top$$

$$G = (vec(F_1), \ldots, vec(F_m))$$

$$h = vec(F_0)$$

and the new A matrix A^{new} is given by,

$$A^{new} = \begin{pmatrix} A \\ G \end{pmatrix}, \quad b^{new} = \begin{pmatrix} b \\ h \end{pmatrix} .$$

Example

```r
library(scs)
A <- matrix(c(1, 1), ncol=1)
b <- c(1, 1)
obj <- 1
cone <- list(f = 2)
control <- list(eps = 1e-3, max_iters = 50)
sol <- scs(A, b, obj, cone, control)
sol
```

Reference

Brendan O'Donoghue, Eric Chu, Neal Parikh, and Stephen Boyd (2013). **"Conic optimization via operator splitting and homogeneous self-dual embedding"** URL http://arxiv.org/abs/1312.3039