CLSOCP: An R package for second order cone programming

Jason Rudy

April 17, 2012

1 Introduction

This document explains the general idea behind the CLSOCP package and gives an example of how to formulate a problem. For details on the optional arguments and output, see the documentation. CLSOCP exports one function, socp, which can be used to solve second order cone programs (SOCPs). An SOCP is an optimization problem of the form

$$\begin{array}{lrl} \min & \sum_{i=1}^{d_q} \langle C_i^q, X_i^q \rangle + \sum_{i=1}^{d_l} \langle C_i^l, X_i^l \rangle , & \\ s.t. & \sum_{i=1}^{d_q} \left\langle A_{ij}^q, X_i^q \right\rangle + \sum_{i=1}^{d_l} \left\langle A_{ij}^l, X_i^l \right\rangle & = b_j, \forall j \in \{1 \dots m\}, \\ & X_i^q & \in Q_{n_i}, \forall i \in \{1 \dots d_q\}, \\ & X_i^l & \in \mathbb{R}_+^{p_j}, \forall i \in \{1 \dots d_l\}, \end{array} \tag{1}$$

where $\langle \cdot, \cdot \rangle$ represents the inner product, Q_n is the second order cone of dimension n, and $\mathbb{R}_+^p$ is the positive orthant of $\mathbb{R}^p$. The A_{ij}^q are referred to as second order cone constraints and the A_{ij}^l are referred to as linear constraints.

The CLSOCP package uses a smoothing Newton method algorithm developed by Xiaoni Chi and Sanyang Liu (2009). The published specification of the Chi-Liu (CL) algorithm includes only the case of $d_q = 1$ and does not include linear constraints ($d_l = 0$). In the CLSOCP package, that algorithm is slightly extended to allow for any $d_q \geq 0$ and $d_l \geq 0$ with $d_q + d_l \geq 1$. The generalized smoothed Fischer-Burmeister function for $d_q > 1$ is the column vector-valued function given by

$$\phi_{general}\left(X_\cdot^q, S_\cdot^q, \mu_\cdot\right) = \begin{pmatrix} \phi\left(X_1^q, S_1^q, \mu_1\right) \\ \vdots \\ \phi\left(X_{d_q}^q, S_{d_q}^q, \mu_{d_q}\right) \end{pmatrix}, \tag{2}$$

where $\phi(X, S, \mu)$ is the smoothed Fischer-Burmeister function (for $d_q = 1$) used by Chi and Liu. The solution to the SOCP coincides with the solution to $\phi_{general}\left(X_\cdot^q, S_\cdot^q, \mu_\cdot\right) = \mathbf{0}$, and the generalized CL algorithm finds the solution to the latter by the Newton method. Linear constraints are accommodated simply by internal conversion to second order cone constraints using the identity $\mathbb{R}^n_+ = \prod_{i=1}^n Q_1$, where $\prod$ here represents the Cartesian product. In all other respects, CLSOCP implements the CL algorithm exactly as originally defined by Chi and Liu.

2 Using the Package

At minimum, five arguments are necessary to use the socp function. To show how to formulate those arguments, let's assume an example with $d_q = 2$, $d_l = 2$, and $m = 3$. The following arguments are required:

A A matrix of dimension m by $\sum_{i=1}^{d_q} n_i + \sum_{i=1}^{d_l} p_i$ formed from the A_i^q and A_i^l such that the constraints in (1) can be stated as $Ax = b$. For our example,

$$A = \begin{bmatrix} A^q_{11} & A^q_{21} & A^l_{11} & A^l_{21} \\ A^q_{12} & A^q_{22} & A^l_{12} & A^l_{22} \\ A^q_{13} & A^q_{23} & A^l_{13} & A^l_{23} \end{bmatrix}, \tag{3}$$

where the A^q_{ij} and A^l_{ij} are row vectors, would be a reasonable formulation. This formulation implicitly defines the solution x as,

$$x = \begin{bmatrix} X^q_1 \\ X^q_2 \\ X^l_1 \\ X^l_2 \end{bmatrix}, \tag{4}$$

where the X^q_i and X^l_i are column vectors, but note that this is not the only possible formulation.

b A vector of length m formed from the b_i such that the constraints in (1) can be stated as $Ax = b$ for the same x as above. For our example,

$$b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{5}$$

c A vector of length $\sum_{i=1}^{d_q} n_i + \sum_{i=1}^{d_l} p_i$ formed from the C^q_i and C^l_i such that the objective objective function in (1) can be stated as $\min \langle c, x\rangle$ for the same x as above. For our example,

$$c = \begin{bmatrix} C^q_1 \\ C^q_2 \\ C^l_1 \\ C^l_2 \end{bmatrix} \tag{6}$$

kvec A vector giving the lengths of the second order cone and linear variables. For our example,

$$kvec = \begin{bmatrix} n_1 \\ n_2 \\ p_1 \\ p_2 \end{bmatrix} \tag{7}$$

type A character vector of the same length as *kvec*, giving the type of the corresponding variable as either *"q"* for second order cone constraints or *"l"* for linear constraints. For our example,

$$type = \begin{bmatrix} "q" \\ "q" \\ "l" \\ "l" \end{bmatrix} \tag{8}$$

Additional arguments can be used to control the behavior of the algorithm, but do not affect the formulation of the problem. For a complete list of optional parameters, as well as the format of the output, see the socp documentation, available from your R command line through the command ?socp.

References

Chi, X. & Liu, S. (2009). A one-step smoothing newton method for second-order cone programming. *Journal of Computational and Applied Mathematics*, 223(1), 114–123.