On Recent Improvements in the Interior-Point Optimizer in MOSEK

ISMP2015 – 14 July 2015 Pittsburgh (US)

Andrea Cassioli, PhD
andrea.cassioli@mosek.com

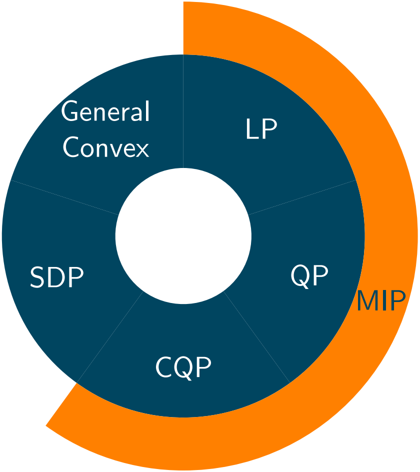
www.mosek.com

Overview

1. Few words about **MOSEK**
2. New features in upcoming v8
3. QCQP to COP automatic conversion
4. Pitfalls in PSD detection
5. Some computational experience

Few words about **MOSEK**

MOSEK is one of the leading provider of high-quality optimization software world-wide.



Version 8 - work in progress

1. Improved presolve.
   - Faster.
   - Eliminator uses much less space.
   - Eliminator has increased stability emphasis.
   - Added some conic presolve.
2. Revised scaling procedure for conic problems:
   - Emphasize accuracy of the unscaled solution.
   - Scales semidefinite problems too.
3. Automatic dualizer for conic problems (no matrix variables).
4. Rewritten interior-point optimizer for conic problems.
   - Emphasize numerical stability for semidefinite problems.
5. QCQPs internally reformulated to conic form.

Convex Quadratic vs. Conic Quadratic

From our practical experience the conic model is :

- numerically more robust,
- easier to exploit duality,
- better when quadratic constraints are present,
- better for primal infeasible problems,
- a more general framework.

However, users are still very much used to QCQPs formulations, therefore

- Convert (QO) to conic form (CQO).
- Map the primal and dual solutions back.

From QCQP to CQO

The quadratic optimization model

$$
\begin{array}{lllll}
\text{minimize} & \frac{1}{2}x^T Q_0^T x + c^T x & & & \\
\text{subject to} & \frac{1}{2}x^T Q_i^T x + a_{i:}x & \leq & b_i, & i = 1\ldots, m. \quad \text{(QO)}
\end{array}
$$

Assumptions:

- Symmetry: $Q_i = Q_i^T, \quad i =, \ldots, m.$
- Convexity: $Q_i \succeq 0.$

Hence, Q_i should be **positive semidefinite.**

The conic optimization model

$$\begin{array}{llll} \text{minimize} & c^T x & & \\ \text{subject to} & Ax & = & b_i, \quad i = 1, \ldots, m, \quad \text{(CQO)} \\ & x \in \mathcal{K}, & & \end{array}$$

where

$$\mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2 \times \cdots .$$

Each $\mathcal{K}_k$ can have the form

- Linear: $\{x \in \mathbb{R}^{n_i} \mid x \geq 0\}$.
- Quadratic: $\{x \in \mathbb{R}^{n_i} \mid x_1 \geq \|x_{2:n_i}\|\}$.
- Rotated quadratic: $\{x \in \mathbb{R}^{n_i} \mid 2x_1 x_2 \geq \|x_{3:n_i}\|^2, \quad x_1, x_2 \geq 0\}$.

The separable reformulation

If $L_i s$ such that $L_i L_i^T = Q_i$ are known, then the separable equivalent is

$$\begin{array}{lrcll} \text{minimize} & \frac{1}{2} f_0^T f_0 + c^T x & & & \\ \text{subject to} & \frac{1}{2} f_i^T f_i + a_{i:} x & \leq & b_i, & i = 1, \ldots, m, \quad \text{(SQO)} \\ & L_i^T x - f_i & = & 0. & \end{array}$$

- The separable problem formulation is (much) bigger.
- But the sparse representation may require much less storage if Q_i is dense but low rank.
- L_i does not have to be lower triangular.

Conic reformulation

From (QO) to (CQO):

$$
\begin{array}{llll}
\text{minimize} & t_0 + c^T x & & \\
\text{subject to} & t_i + a_{i:} x & = & b_i, \quad i =, 1 \ldots, m, \quad \text{(CQO)} \\
& L_i^T x - f_i & = & 0, \\
& z_i & = & 1, \\
& 2 z_i t_i \geq \left\| f_i \right\|^2 . & &
\end{array}
$$

- Theory:
  - Both problems solves in the same worst case complexity using an interior-point method.
  - No bad duality states is introduced in the conic reformulation ART [1].

Conic Reformulation

Converting QO to CQO is a trivial procedure once L_i's are known.
So who should do that?

the user!

- Factorization may be already available.
- Better control on the choice of the way to factorize Q_i's,

However, **MOSEK** v8 will make the conversion automatically.

Quadratic PSD form characterization

The statements are equivalent

$$
\begin{array}{ll}
i) & Q_i \succeq 0. \\
ii) & \lambda_{\min}(Q_i) \geq 0. \\
iii) & \exists L_i \mid Q_i = L_i L_i^T. \\
iv) & v^T Q_i v \geq 0, \quad \forall v.
\end{array}
$$

Practical observation:

- How does the modeler knows (QO) is convex?
- Claim: The modeler knows L_i!

Automatic conversion implemented in **MOSEK** (I)

Purpose is to compute L such that

$$Q = LL^T$$

or in practice

$$Q \approx LL^T$$

considering rounding errors.

Assumptions on the users:

- Users applies this to (near) positive semidefinite problems.
- Users prefer a false positive to a false negative.

How to deal with factorizations?

Motivating example

$$\begin{array}{lll} \text{minimize} & -x_1 - x_2 & \\ \text{subject to} & (x_1 - x_2)^2 & \leq \; 0, \\ & 0 \leq x_1, x_2 \leq 1 & \end{array}$$

Often in practice the quadratic constraints could be affected by a small error ε, i.e.

$$x^T \begin{bmatrix} 1 & -1 \\ -1 & 1+\epsilon \end{bmatrix} x \leq 0$$

Typical error sources:

- Introduced by user.
- Coming from finite precision floating point precision computations.

Practicabilities about the conversion

Observe:

- $\epsilon < 0$: The problem is not convex.
- $\epsilon = 0$: $x_1^* = x_2^* = 1$.
- $\epsilon > 0$: $x_1^* = x_2^* = 0$.

Conclusions:

- Hard to produce a 100% automatic fool proof conversion.
- Conversion should be done at the modelling stage!

Automatic conversion implemented in **MOSEK** (II)

Lemma

If Q is symmetric positive semidefinite then it holds

$$e_1^T Q e_1 = Q_{11} \geq 0$$

and

$$Q_{11} = 0 \Rightarrow Q_{1:} = Q_{:1} = 0.$$

Automatic conversion implemented in **MOSEK** (III)

Lemma

If Q is symmetric positive semidefinite and $Q_{11} > 0$, then

$$
\begin{aligned}
Q &= E_1 Q_1 E_1^T \\
Q_1 &= \begin{bmatrix} 1 & 0 \\ 0 & Q_{22} - \dfrac{Q_{21} Q_{21}^T}{Q_{11}} \end{bmatrix}
\end{aligned}
$$

where

$$
E = \begin{bmatrix} \sqrt{Q_{11}} & 0 \\ Q_{21}/\sqrt{Q_{11}} & I \end{bmatrix}.
$$

Moreover,

$$
Q_{22} - \frac{Q_{21} Q_{21}^T}{Q_{11}}
$$

will be positive semidefinite.

Automatic conversion implemented in **MOSEK** (IV)

Hence, if Q is positive definite then

$$Q = LL^T$$

where

$$L = E_1 E_2 \cdots E_n.$$

Fact: L will be lower triangular.
But what if

$$Q_{11} \approx 0?$$

Automatic conversion implemented in **MOSEK** (V)

- $Q_{11} \leq -\varepsilon$ then Q is said to be NOT positive semidefinite.
- $-\varepsilon < Q_{11} \leq \varepsilon$ then
  - Replace Q_{11} by ε.
  - If the complete Q is determined PSD, then replace $L_{:1}$ by 0 in the final result.
- Default value: $\varepsilon = 10^{-10}$.

The procedure will detect

$$\begin{bmatrix} 0 & 1 \\ 1 & 10^8 \end{bmatrix}$$

negative semidefinite.

Automatic conversion implemented in **MOSEK** (VI)

Note the procedure is applied to a scaled Q i.e.

$$SQS^T$$

where $S = diag(s)$ and all diagonal elements of SQS^T belongs to $\{-1, 0, 1\}$. Makes the usage of a absolute constant sensible.

MOSEK results

The **MOSEK** procedure produces on our example:

$$L = \begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix}.$$

An alternative procedure

- $Q_{11} \leq -\varepsilon$ then Q is said to be NOT positive semidefinite.
- $-\varepsilon < Q_{11} \leq \varepsilon$ then replace Q_{11} by ε.

Take a look at the example

$$Q = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

and hence

$$L = \begin{bmatrix} 1 & 0 \\ -1 & 10^{-10} \end{bmatrix}$$

which most likely is not what the user intended because this implies $x = 0$.

Discussion

- Procedure can be fooled.
- Alternative approaches:
  - Revised Schnable and Eskow approach [5].
  - Rank revealing Cholesky [4]. (Pivotting required!)
- Alternatives are computational more complicated or (much more) expensive.

Preliminary computation results

Setting:

- 64 bit Linux.
- 1 thread only.
- v7.1 vs. v8
- Public and customer provided models.

	time
Small	$\leq 6s$
Medium	$\leq 60s$
Large	$> 60s$

An optimizer o is declared a winner if

$$t_o \leq \max(t_{\min} + 0.01, 1.005 t_{\min}).$$

Algorithms in MOSEK

- (QO): Solves a homogenized KKT system using (=nonsymmetric primal-dual algorithm) ([3]).

- (CQO): Symmetric primal-dual algorithm based on the Nesterov-Todd direction ART ([2]).

Quadratic problems (linear constraints only)

	small		medium		large	
	7.1	8.0	7.1	8.0	7.1	8.0
Num.	220	220	10	10	1	1
Firsts	187	158	2	8	0	1
Total time	128.41	56.20	359.13	311.56	444.28	244.01

Param ILS instances

Available at www.cs.ubc.ca/labs/beta/Projects/ParamILS/.

	7.1	8.0
Num.	100	100
Firsts	0	100
Total time	917.955	90.179

Quadratically constrained problems

	small		medium	
	7.1	8.0	7.1	8.0
Num.	239	239	8	8
Firsts	161	150	3	5
Total time	350.790	94.290	1360.417	213.454

Discussion

- Conic reformulations wins because
  - it requires less iterations.
  - dualization sometimes lead to huge wins.
  - employs better linear algebra (newer code path).

However, for smallish models the nonconic formulation is better.

Summary

- **MOSEK** version 8 will internally solve quadratic and quadratically constrained problems on conic form.
  - Improves robustness,
  - Solution speed on average.
- Checking positive semi definiteness is tricky.
  - It is recommended to formulate problem on conic form
  - or as a separable problem.

Thank you!

Andrea Cassioli, PhD
andrea.cassioli@mosek.com

www.mosek.com

References

E. D. Andersen, C. Roos, and T. Terlaky.
Notes on duality in second order and p -order cone optimization.
Optimization, 51(4):627–643, 2002.

E. D. Andersen, C. Roos, and T. Terlaky.
On implementing a primal-dual interior-point method for conic quadratic optimization.
Math. Programming, 95(2), February 2003.

E. D. Andersen and Y. Ye.
On a homogeneous algorithm for the monotone complementarity problem.
Math. Programming, 84(2):375–399, February 1999.

M. Gu and L. Miranian.
Strong rank revealing cholesky factorization.
Electronic Transactions on Numerical Analysis, 17:76–92, 2004.

R. B. Schnable and E. Eskow.
A revised modified Cholesky Factorization Algorithm.
SIAM J. on Optim., 9(4):1135–1148, 1999.