



## Optimization in finance with MOSEK

April 22, 2017

Yinyu Ye  
(Erling D. Andersen)

[www.mosek.com](http://www.mosek.com)



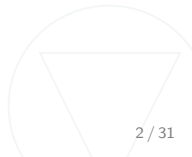
## Section 1

### Background



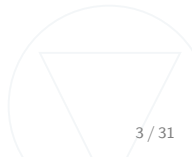


- MOSEK is a Danish company.
- Vision: Create and sell software for mathematical optimization problems.
  - Linear and conic problems.
  - Convex problems.
  - Some integer problems.
- Located in Copenhagen Denmark at Symbion Science Park.
- Daily management: Erling D. Andersen.
- Chairman of technical advisory board: Yinyu Ye





- The original Markowitz optimization model in Portfolio Management
- The distributionally robust and data-driven portfolio management
- The portfolio management with regulations and trading cost control
- Trading model: optimal hedge/unwinding strategy (trade-off between trading cost and risk)
- Risk management such as Risk/VarR minimization



## Section 2

### Product overview





- Main product: MOSEK optimization suite.
- Solves generic:
  - Linear, quadratic, and nonlinear problems [3, 2]. Only convex cases.
  - Conic optimization problems.

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b, \\ & x \in \mathcal{K} \end{array}$$

where  $K$  is a convex cone (only linear, quadratic, semidefinite).

- Integer optimization problems.
  - Same as above but some variables are integer constrained.
- The software is **NOT** application specific but very suitable in finance.

## Section 3

### Conic optimization

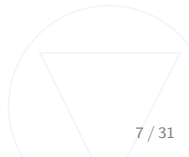




$$\begin{aligned} & \text{minimize} && \sum (c^k)^T x^k \\ & \text{subject to} && \sum_k A^k x^k = b, \\ & && x^k \in \mathcal{K}^k, \quad \forall k, \end{aligned}$$

where

- $c^k \in \mathbb{R}^{n^k}$ ,
- $A^k \in \mathbb{R}^{m \times n^k}$ ,
- $b \in \mathbb{R}^m$ ,
- $\mathcal{K}^k$  are convex cones.





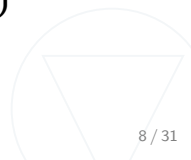


The linear cone:

$$\{x \in \mathbb{R} : x \geq 0\}.$$

The quadratic cones:

$$\mathcal{K}_q := \left\{ x \in \mathbb{R}^n : x_1 \geq \sqrt{\sum_{j=2}^n x_j^2} \right\},$$
$$\mathcal{K}_r := \left\{ x \in \mathbb{R}^n : 2x_1x_2 \geq \sum_{j=3}^n x_j^2, x_1, x_2 \geq 0 \right\}.$$



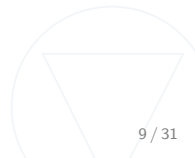


The cone of symmetric positive semidefinite matrices:

$$\mathcal{K}_s := \{X \in \mathbb{R}^{n \times n} \mid X = X^T, \lambda_{\min}(X) \geq 0\}$$

Facts:

- The 3 cones belong to the class of symmetric cones.
- Only 5 different symmetric cones.





Exponential cone:

$$\mathcal{K}_e := cl \left\{ x \in \mathbb{R}^n : x_1 \geq x_2 e^{x_3/x_2}, x_2 > 0 \right\}.$$

Power cone:

$$\mathcal{K}_p(\alpha) := \left\{ x \in \mathbb{R}^n : \prod_{j=1}^{n_1} x_j^{\alpha_j} \geq \|x_{n_1+1:n}\|, x_{1:n_1} \geq 0 \right\}$$

where

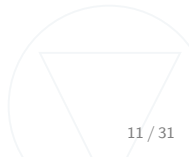
$$\sum_{j=1}^{n_1} \alpha_j = 1 \text{ and } \alpha_j \geq 0.$$

- MOSEK cannot handle nonsymmetric cones yet.
- However, work in progress based on [7, 1] and others.
- No promises!





- Model is by construction convex which is **important**.
- Model can be specified using simple data. No  $f(x)$  and derivatives.
- Almost any convex problem can be modeled.
  - Nemirovski [6].
  - Lubin et al. [5] shows all 333 instances in MINLPLIB2 is conic representable
- Powerful primal-dual alg.s exist for **symmetric cones**.
- Leads to **Extreme disciplined optimization** inspired by [4].
- Almost all convex problems can be specified in this framework.
- Please informs us about examples that cannot.



## Section 4

Finance application: Portfolio optimization





An investor can invest in  $n$  stocks or assets to be held over a period of time. What is the optimal portfolio?

Now assume a stochastic model where the return of the assets is a random variable

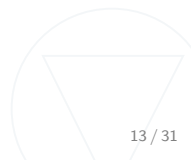
$$r$$

with known mean

$$\mu = \mathbf{E}r$$

and covariance

$$\Sigma = \mathbf{E}(r - \mu)(r - \mu)^T.$$





Let  $x_j$  be the amount invested in asset  $j$ . Moreover, the expected return is:

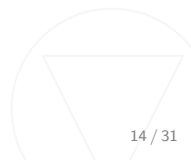
$$\mathbf{E}y = \mu^T x$$

and variance of the return is

$$(y - \mathbf{E}y)^2 = x^T \Sigma x.$$

The investors optimization problem:

$$\begin{array}{ll} \text{maximize} & \mu^T x \\ \text{subject to} & e^T x = w + e^T x^0, \\ & x^T \Sigma x \leq \gamma^2, \\ & x \geq 0, \end{array}$$



where

- $e$  is the vector of all ones.
- $w$  investors initial wealth.
- $x^0$  investors initial portfolio.
- Objective maximize expected return.
- Constraints:
  - Budget constraint.  $(e^T x = \sum_{j=1}^n x_j)$ .
  - Risk constraint.  $\gamma$  is chosen by the investor.
  - Only buy a positive amount i.e. no short-selling.





The covariance matrix  $\Sigma$  is positive semidefinite by definition.  
Therefore,

$$\exists G : \quad \Sigma = GG^T.$$

CQ reformulation:

$$\begin{array}{ll} \text{maximize} & \mu^T x \\ \text{subject to} & e^T x = w + e^T x^0, \\ & [\gamma; G^T x] \in \mathcal{K}_q^{n+1}, \\ & x \geq 0. \end{array}$$

because

$$[\gamma; G^T x] \in \mathcal{K}_q^{n+1} \Rightarrow \gamma \geq \|G^T x\| \Rightarrow \gamma^2 \geq x^T GG^T x.$$





```
import mosek
import sys

from mosek.fusion import *
from portfolio_data import *

def BasicMarkowitz(n,mu,GT,x0,w,gamma):
    with Model("Basic Markowitz") as M:

        # Redirect log output from the solver to stdout for debugging.
        # if uncommented.
        M.setLogHandler(sys.stdout)

        # Defines the variables (holdings). Shortselling is not allowed.
        x = M.variable("x", n, Domain.greaterThan(0.0))

        # Maximize expected return
        M.objective('obj', ObjectiveSense.Maximize, Expr.dot(mu,x))

        # The amount invested must be identical to initial wealth
        M.constraint('budget', Expr.sum(x), Domain.equalsTo(w+sum(x0)))

        # Imposes a bound on the risk
        M.constraint('risk', Expr.vstack( gamma,Expr.mul(GT,x)), Domain.inQCone())

        M.solve()

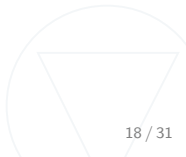
        return (M.primalObjValue(), x.level())

if __name__ == '__main__':

    (expret,x) = BasicMarkowitz(n,mu,GT,x0,w,gamma)
    print("Expected return: %e" % expret)
    print("x: "),
    print(x)
```



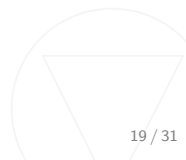
```
n      = 3;
w      = 1.0;
mu     = [0.1073,0.0737,0.0627]
x0     = [0.0,0.0,0.0]
gamma  = 0.04
GT     = [[ 0.166673333200005, 0.0232190712557243 , 0.0012599496030238 ],
          [ 0.0                , 0.102863378954911 , -0.00222873156550421],
          [ 0.0                , 0.0                , 0.0338148677744977 ]]
```





Running

```
python portfolio_basic.py
```





```

Optimizer - threads                : 4
Optimizer - solved problem         : the primal
Optimizer - Constraints             : 3
Optimizer - Cones                  : 1
Optimizer - Scalar variables       : 6
Optimizer - Semi-definite variables: 0
Factor - setup time                : 0.00
Factor - ML order time             : 0.00
Factor - nonzeros before factor    : 6
Factor - dense dim.               : 0
Optimizer - conic                  : 4
Optimizer - scalarized             : 0
Factor - dense det. time          : 0.00
Factor - ML order time            : 0.00
Factor - GP order time            : 0.00
Factor - nonzeros before factor    : 6
Factor - after factor              : 6
Factor - flops                    : 7.00e+001
ITE PFEAS DFEAS GFEAS PRSTATUS POBJ DOBJ MU TIME
0  1.0e+000 1.0e+000 1.0e+000 0.00e+000 0.000000000e+000 0.000000000e+000 1.0e+000 0.00
1  1.7e-001 1.7e-001 4.4e-001 9.46e-001 1.259822223e-001 2.171837612e-001 1.7e-001 0.00
2  4.0e-002 4.0e-002 5.6e-001 1.56e+000 8.104070951e-002 1.693911786e-001 4.0e-002 0.00
3  1.4e-002 1.4e-002 2.9e-001 3.00e+000 7.268285567e-002 8.146211968e-002 1.4e-002 0.00
4  1.3e-003 1.3e-003 1.1e-001 1.43e+000 7.102726686e-002 7.178857777e-002 1.3e-003 0.00
5  1.7e-004 1.7e-004 3.9e-002 1.05e+000 7.101472221e-002 7.111329525e-002 1.7e-004 0.00
6  7.7e-006 7.7e-006 8.5e-003 1.01e+000 7.099770619e-002 7.100232290e-002 7.7e-006 0.00
7  6.0e-007 6.0e-007 2.4e-003 1.00e+000 7.099794084e-002 7.099830405e-002 6.0e-007 0.00
8  1.7e-008 1.7e-008 4.0e-004 1.00e+000 7.099799652e-002 7.099800667e-002 1.7e-008 0.00
Interior-point optimizer terminated. Time: 0.00.

```

Optimizer terminated. Time: 0.01

Expected return: 7.099800e-02

x:

[ 0.15518625 0.12515363 0.71966011]

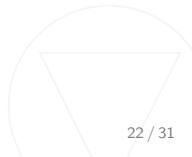
## Section 5

### Software overview





- Optimizer API
  - Matrix orientated.
  - C, Java, .NET and Python.
  - Low computational over and high flexibility.
  - Somewhat cumbersome.
- Fusion API
  - Object orientated.
  - Only linear and conic problems.
  - C++, Java, MATLAB, .NET and Python.



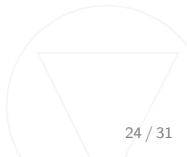


- Optimization server.
- Other
  - AMPL, GAMS.
  - MATLAB toolbox.
  - R package.
  - Julia, Go.
  - File base: CBF, OPF, MPS.
- Third party links:
  - Commercial: CVX (MATLAB), GAMS.
  - Opens source: CVXPY, Yalmip (MATLAB) and more.





- Handles large scale sparse case.
- Algorithms: Simplex and interior-point methods.
- Efficiency
  - Code in C.
  - Exploits density and sparsity.
  - Exploit modern hardware through the usage of the Intel MKL BLAS library.
  - Is multi threaded i.e. exploit multiple CPU cores.



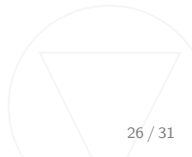


- Sparse Cholesky computation.
- Some BLAS like routines available e.g. dense matrix products.
- Infeasibility analysis.
- Problem analysis of scaling and sparsity.
- Automatic and transparent dualizer.
- Automatic scaling.





- Documentation at `https://mosek.com/resources/doc`
  - Manuals for interfaces.
  - Modelling cook book.
  - White papers.
- Examples
  - Tutorials at Github:  
`https://github.com/MOSEK/Tutorials`



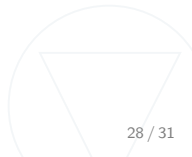
## Section 6

### Summary



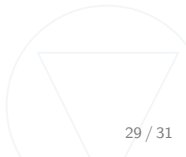


- Conic (and convex) optimization is useful in finance applications.
- MOSEK is a software package for (mixed integer) conic (and convex) optimization
- MOSEK is very useful in finance.



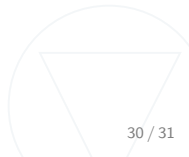


- [1] A homogeneous interior-point algorithm for nonsymmetric convex conic optimization.  
*Math. Programming*, 150:391–422, May 2015.
- [2] E. D. Andersen and Y. Ye.  
A computational study of the homogeneous algorithm for large-scale convex optimization.  
*Computational Optimization and Applications*, 10:243–269, 1998.
- [3] E. D. Andersen and Y. Ye.  
On a homogeneous algorithm for the monotone complementarity problem.  
*Math. Programming*, 84(2):375–399, February 1999.





- [4] M. Grant, S. Boyd, and Y. Ye.  
Disciplined convex programming.  
In L. Liberti and N. Maculan, editors, *Global Optimization: From Theory to Implementation*.
- [5] Miles Lubin and Emre Yamangil and Russell Bent and Juan Pablo Vielma.  
Extended Formulations in Mixed-integer Convex Programming.  
Technical report, 2016.  
[arXiv:1607.03566v1](https://arxiv.org/abs/1607.03566v1).





- [6] A. Nemirovski.  
Advances in convex optimization: Conic programming.  
In Marta Sanz-Sol, Javier Soria, Juan L. Varona, and Joan Verdera, editors, *Proceedings of International Congress of Mathematicians, Madrid, August 22-30, 2006, Volume 1*, pages 413–444. EMS - European Mathematical Society Publishing House, April 2007.
- [7] Santiago Akle Serrano.  
*Algorithms for unsymmetric cone optimization and an implementation for problems with the exponential cone.*  
PhD thesis, Stanford University, 2015.

