The R-to-MOSEK Optimization Interface

Henrik Alsing Friberg

MOSEK ApS,
Fruebjergvej 3, Box 16, DK 2100 Copenhagen,
Blog: http://themosekblog.blogspot.com

ISMP, Berlin 2012

http://www.mosek.com

Introduction

What is R?

The R Project

A fast open-source programming language for technical computing and graphics.

Highlights:

- One million users – Intel Capital, 2009
- The Comprehensive R Archive Network (3895 packages and counting)
- Direct interfaces to C and Fortran (as well as C++, Perl, Java, Python, Matlab, Excel etc.)
- Revolution Analytics
- RStudio

What is MOSEK?

MOSEK

A high performance software library designed to solve large-scale convex optimization problems.

Highlights:

- Linear constraints (LP)
- Second-order cone constraints (SOCP)
- Semidefinite constraints (SDP)
- Mixed-integer variables (MIP)
- Quadratic terms (QP and QCQP)
- Separable convex operators – $\log(x_i)$, $\exp(x_i)$, ...

What is Rmosek?

Rmosek

A simple interface from R to the optimizers of MOSEK.

Why?

- Some financial customers find R better than MATLAB.
- Replace unofficial implementations.

http://www.mosek.com

What is Rmosek?

Rmosek

A simple interface from R to the optimizers of MOSEK.

Why?

- Some financial customers find R better than MATLAB.
- Replace unofficial implementations.

Goals:

- Open-source
- Highly effective
- Integrate with what R users normally do

Installation

A oneliner for Windows, MacOS and Linux:

```
install.packages("Rmosek", type="source")
```

Installation

A oneliner for Windows, MacOS and Linux:

```
install.packages("Rmosek", type="source")
```

Package overview:

```
library(help="Rmosek")
```

Major functions:

- `mosek(prob)`
- `mosek_write(prob, file)`
- `mosek_read(file)`

The problem is an R-variable.. Create as you like!

Simple models

Linear optimization

$$
\begin{array}{lrcrcrcrcl}
\text{maximize} & 3x_1 & + & 1x_2 & + & 5x_3 & + & 1x_4 & & \\
\text{subject to} & 3x_1 & + & 1x_2 & + & 2x_3 & & & = & 30, \\
 & 2x_1 & + & 1x_2 & + & 3x_3 & + & 1x_4 & \geq & 15, \\
 & & & 2x_2 & & & + & 3x_4 & \leq & 25,
\end{array}
$$

$$
\begin{array}{ccccc}
0 & \leq & x_1 & \leq & \infty, \\
0 & \leq & x_2 & \leq & 10, \\
0 & \leq & x_3 & \leq & \infty, \\
0 & \leq & x_4 & \leq & \infty.
\end{array}
$$

Linear optimization


```
lo1 <- list()

## Objective
lo1$sense <- "max"
lo1$c     <- c(3, 1, 5, 1)

## Constraint matrix
lo1$A <- spMatrix(nrow = 3, ncol = 4,
  i = c(1, 2, 1, 2, 3, 1, 2, 2, 3),
  j = c(1, 1, 2, 2, 2, 3, 3, 4, 4),
  x = c(3, 2, 1, 1, 2, 2, 3, 1, 3)
)
```

Linear optimization

```
## Constraint bounds
lo1$bc <- cbind(c(30, 30),
                c(15, Inf),
                c(-Inf, 25))

## Variable bounds
lo1$bx <- cbind(c(0, Inf),
                c(0, 10),
                c(0, Inf),
                c(0, Inf))
```

Advanced models

Second-order cone and semidefinite optimization

$$
\begin{array}{lll}
\text{minimize} & \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix} \bullet \overline{x}_1 + x_1 & \\
\text{subject to} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \overline{x}_1 + x_1 & = \quad 1.0\,, \\
& \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \bullet \overline{x}_1 + x_2 + x_3 & = \quad 0.5\,,
\end{array}
$$

$$
(x_1, x_2, x_3) \in \mathbb{R}^3, \quad x_1 \geq \sqrt{x_2^2 + x_3^2}, \quad \overline{x}_1 \succeq 0\,.
$$

Second-order cone and semidefinite optimization

$$
\begin{array}{ll}
\text{minimize} & \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix} \bullet \overline{x}_1 + \mathbf{x_1} \\
\text{subject to} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \overline{x}_1 + x_1 \qquad = \quad 1.0\,, \\
& \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \bullet \overline{x}_1 + x_2 + x_3 \quad = \quad 0.5\,, \\
& (x_1, x_2, x_3) \in \mathbb{R}^3, \quad x_1 \geq \sqrt{x_2^2 + x_3^2}, \quad \overline{x}_1 \succeq 0\,.
\end{array}
$$

```
sdo1 <- list()

sdo1$sense <- "minimize"
sdo1$c     <- c(1,0,0)
```

Second-order cone and semidefinite optimization

$$
\begin{array}{lll}
\text{minimize} & \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix} \bullet \overline{x}_1 + x_1 & \\
\text{subject to} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \overline{x}_1 + \mathbf{x_1} & = \; 1.0\,, \\
& \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \bullet \overline{x}_1 + \mathbf{x_2} + \mathbf{x_3} & = \; 0.5\,,
\end{array}
$$

$$(x_1, x_2, x_3) \in \mathbb{R}^3, \quad x_1 \geq \sqrt{x_2^2 + x_3^2}, \quad \overline{x}_1 \succeq 0\,.$$

```
sdo1$A <- spMatrix(nrow=2, ncol=3,
    i = c(1,2,2),
    j = c(1,2,3),
    x = c(1,1,1)
)
```

Second-order cone and semidefinite optimization

$$
\begin{array}{ll}
\text{minimize} & \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix} \bullet \overline{x}_1 + x_1 \\
\text{subject to} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \overline{x}_1 + x_1 \qquad = \quad \mathbf{1.0}\,, \\
& \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \bullet \overline{x}_1 + x_2 + x_3 \quad = \quad \mathbf{0.5}\,,
\end{array}
$$

$$(x_1, x_2, x_3) \in \mathbb{R}^3, \quad x_1 \geq \sqrt{x_2^2 + x_3^2}, \quad \overline{x}_1 \succeq 0\,.$$

```
sdo1$bc <- cbind(c(1.0, 1.0),
                 c(0.5, 0.5))
```

Second-order cone and semidefinite optimization

$$\begin{array}{ll} \text{minimize} & \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix} \bullet \overline{x}_1 + x_1 \\ \text{subject to} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \overline{x}_1 + x_1 \quad = \quad 1.0\,, \\ & \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \bullet \overline{x}_1 + x_2 + x_3 \quad = \quad 0.5\,, \end{array}$$

$$(\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}) \in \mathbb{R}^{\mathbf{3}}, \quad x_1 \geq \sqrt{x_2^2 + x_3^2}, \quad \overline{x}_1 \succeq 0\,.$$

```
sdo1$bx <- cbind(c(-Inf, Inf),
                 c(-Inf, Inf),
                 c(-Inf, Inf))
```

Second-order cone and semidefinite optimization

$$
\begin{aligned}
\text{minimize} \quad & \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix} \bullet \overline{x}_1 + x_1 \\
\text{subject to} \quad & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \overline{x}_1 + x_1 \quad = \quad 1.0\,, \\
& \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \bullet \overline{x}_1 + x_2 + x_3 \quad = \quad 0.5\,, \\
& (x_1, x_2, x_3) \in \mathbb{R}^3, \quad \mathbf{x_1} \geq \sqrt{\mathbf{x_2^2} + \mathbf{x_3^2}}, \quad \overline{x}_1 \succeq 0\,.
\end{aligned}
$$

```
sdo1$cones <- cbind(
  list("quad", c(1,2,3))
)
```

Second-order cone and semidefinite optimization

$$\begin{array}{ll} \text{minimize} & \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix} \bullet \overline{x}_1 + x_1 \\ \text{subject to} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \overline{x}_1 + x_1 \quad = \quad 1.0\,, \\ & \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \bullet \overline{x}_1 + x_2 + x_3 \quad = \quad 0.5\,, \end{array}$$

$$(x_1, x_2, x_3) \in \mathbb{R}^3, \quad x_1 \geq \sqrt{x_2^2 + x_3^2}, \quad \overline{\mathbf{x}}_1 \succeq \mathbf{0}\,.$$

```
sdo1$bardim <- c(3)
```

Second-order cone and semidefinite optimization

$$\begin{array}{ll} \text{minimize} & \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix} \bullet \overline{\mathbf{x}}_{\mathbf{1}} + x_1 \\ \text{subject to} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \overline{x}_1 + x_1 \qquad = \quad 1.0\,, \\ & \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \bullet \overline{x}_1 + x_2 + x_3 \quad = \quad 0.5\,, \end{array}$$

$$(x_1, x_2, x_3) \in \mathbb{R}^3, \quad x_1 \geq \sqrt{x_2^2 + x_3^2}, \quad \overline{x}_1 \succeq 0\,.$$

```
# Block triplet format (SDPA)
sdo1$barc$j <- c(1,1,1,1,1)

sdo1$barc$k <- c(1,2,2,3,3)
sdo1$barc$l <- c(1,1,2,2,3)
sdo1$barc$v <- c(2,1,2,1,2)
```

Second-order cone and semidefinite optimization

$$
\begin{array}{lrcl}
\text{minimize} & \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix} \bullet \overline{x}_1 + x_1 & & \\
\text{subject to} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \overline{\mathbf{x}}_1 + x_1 & = & 1.0\,, \\
& \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \bullet \overline{\mathbf{x}}_1 + x_2 + x_3 & = & 0.5\,,
\end{array}
$$

$$
(x_1, x_2, x_3) \in \mathbb{R}^3, \quad x_1 \geq \sqrt{x_2^2 + x_3^2}, \quad \overline{x}_1 \succeq 0\,.
$$

```
# Block triplet format (SDPA)
sdo1$barA$i <- c(1,1,1, 2,2,2,2,2,2)
sdo1$barA$j <- c(1,1,1, 1,1,1,1,1,1)

sdo1$barA$k <- c(1,2,3, 1,2,3,2,3,3)
sdo1$barA$l <- c(1,2,3, 1,1,1,2,2,3)
sdo1$barA$v <- c(1,1,1, 1,1,1,1,1,1)
```

Better formulations

General quadratic formulation

$$
\begin{array}{ll}
\text{minimize} & \frac{1}{2}x^T Q x + c^T x + e^T v + k \\
\text{subject to} & -v \;\leq\; x \;\leq\; v \,,
\end{array}
$$

```
rr <- mosek_read("probQP.opf")
qp <- rr$prob

rqp <- mosek(qp)
```

General quadratic formulation

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^TQx + c^Tx + e^Tv + k \\ \text{subject to} & -v \leq x \leq v, \end{array}$$

ITE	PFEAS	DFEAS	POBJ	TIME
8	1.3e-14	3.1e-08	4.213156702e+03	74.84
9	1.2e-14	2.5e-09	4.213153627e+03	82.54
10	1.1e-14	1.3e-10	4.213153292e+03	90.21

Separable quadratic formulation

Cholesky Factorization: $Q = A^T A$

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} w^T w + c^T x + e^T v + k \\ \text{subject to} & -v \leq x \leq v, \\ & Ax = w, \end{array}$$

mosek
http://www.mosek.com

Separable quadratic formulation

Cholesky Factorization: $Q = A^T A$

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} w^T w + c^T x + e^T v + k \\ \text{subject to} & -v \le x \le v, \\ & Ax = w, \end{array}$$

```
Q <- sparseMatrix(qp$qobj$i,
                  qp$qobj$j,
                  x = qp$qobj$v,
                  dims = c(nx,nx),
                  symmetric = TRUE)
A <- chol(Q)
```

Separable quadratic formulation

Cholesky Factorization: $Q = A^T A$

$$
\begin{array}{ll}
\text{minimize} & \frac{1}{2} w^T w + c^T x + e^T v + k \\
\text{subject to} & -v \;\leq\; x \;\leq\; v\,, \\
 & \qquad\; Ax \;=\; w\,,
\end{array}
$$

```
sqp <- qp

sqp$qobj <- list(i = 1:nx,
                 j = 1:nx,
                 v = rep(1,nx))
```

Separable quadratic formulation

Cholesky Factorization: $Q = A^T A$

$$
\begin{array}{ll}
\text{minimize} & \frac{1}{2} w^T w + c^T x + e^T v + k \\
\text{subject to} & -v \;\leq\; x \;\leq\; v\,, \\
& \qquad\quad Ax \;=\; w\,,
\end{array}
$$

```
eye   <- Diagonal(nx)
zeros <- Matrix(0, nrow=nx, ncol=nx)

sqp$A  <- rBind(
  sqp$A,
  cBind(-eye, A, zeros)
)
```

Separable quadratic formulation

Cholesky Factorization: $Q = A^T A$

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} w^T w + c^T x + e^T v + k \\ \text{subject to} & -v \leq x \leq v \,, \\ & Ax = w \,, \end{array}$$

ITE	PFEAS	DFEAS	POBJ	TIME
17	1.1e-06	1.5e-08	4.213156615e+03	29.76
18	1.0e-06	1.5e-08	4.213156490e+03	31.36
19	8.0e-06	4.8e-10	4.213153446e+03	32.99

Conic quadratic formulation

$$
\begin{array}{ll}
\text{minimize} & \frac{1}{2}p + c^T x + e^T v + k \\
\text{subject to} & \begin{array}{rcccl} -v & \leq & x & \leq & v\,, \\ & & Ax & = & w\,, \\ & & 2qp & \geq & \|w\|_2^2 \\ & & q & = & \frac{1}{2} \end{array}
\end{array}
$$

Conic quadratic formulation

$$
\begin{array}{ll}
\text{minimize} & \frac{1}{2}p + c^T x + e^T v + k \\
\text{subject to} & \begin{array}{ccccc} -v & \leq & x & \leq & v\,, \\ & & Ax & = & w\,, \\ & & 2qp & \geq & \|w\|_2^2 \\ & & q & = & \frac{1}{2} \end{array}
\end{array}
$$

```
cqp$qobj <- NULL
```

http://www.mosek.com

Conic quadratic formulation

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}p + c^T x + e^T v + k \\ \text{subject to} & \begin{array}{rcccl} -v & \leq & x & \leq & v\,, \\ & & Ax & = & w\,, \\ & & 2qp & \geq & \|w\|_2^2 \\ & & q & = & \frac{1}{2} \end{array} \end{array}$$

```
cqp$c  <- c(0.0, 0.5, cqp$c)

cqp$bx <- cBind( c(0.5, 0.5),
                 c(0.0, Inf),
                 cqp$bx       )
```

Conic quadratic formulation

$$
\begin{array}{ll}
\text{minimize} & \frac{1}{2}p + c^T x + e^T v + k \\
\text{subject to} & \begin{array}{rcccl} -v & \leq & x & \leq & v\,, \\ & & Ax & = & w\,, \\ & & 2qp & \geq & \|w\|_2^2 \\ & & q & = & \frac{1}{2} \end{array}
\end{array}
$$

```
cqp$cones <- cBind(
  list("rquad", 1:(2+nw))
)
```

Conic quadratic formulation

$$
\begin{array}{ll}
\text{minimize} & \frac{1}{2}p + c^T x + e^T v + k \\
\text{subject to} & \begin{array}{ccccc} -v & \leq & x & \leq & v\,, \\ & & Ax & = & w\,, \\ & & 2qp & \geq & \|w\|_2^2 \\ & & q & = & \frac{1}{2} \end{array}
\end{array}
$$

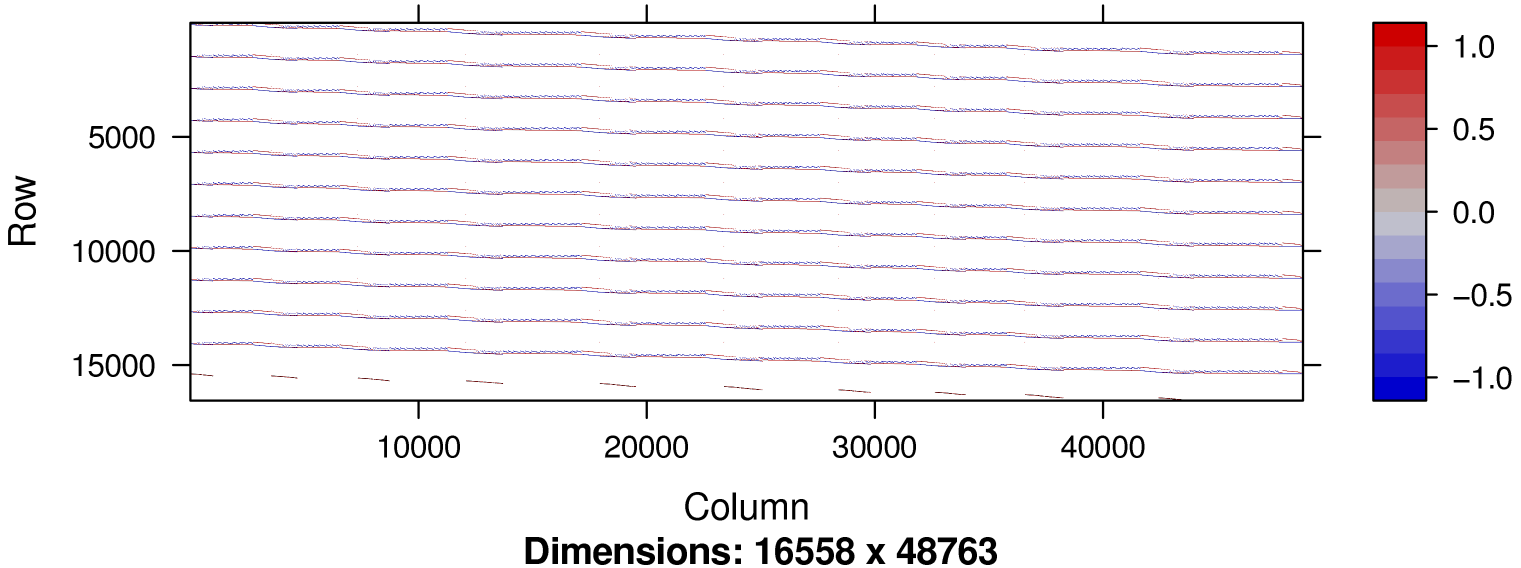
ITE	PFEAS	DFEAS	POBJ	TIME
10	8.7e-07	1.7e-06	4.213161986e+03	6.67
11	1.3e-07	2.5e-07	4.213154532e+03	7.17
12	1.2e-07	2.5e-07	4.213154518e+03	7.66

Data handling and visualization

Problem inspection


```
r <- mosek_read("network.mps")
problem <- r$prob

pdf("picture.pdf")
print(image(problem$A))
dev.off()
```



Math made easy

Normalize a data-matrix:
(each column of X is a time series)

```
N <- nrow(X)
mu_r <- matrix(1,nrow=N) %*% colMeans(X)
Xbar <- 1/sqrt(N-1) * (X - mu_r)
```

Math made easy

Find the QR factorization:

```
factor <- qr(Xbar)
Q <- qr.Q(factor)
R <- qr.R(factor)
```

Math made easy

Save results and read them back in:

```
R <- as.matrix(
  read.csv("data/qr-r.csv", header=FALSE)
)
```

The efficient frontier

$$\begin{array}{lll} \text{minimize} & r^T(w^0+x) - \lambda \|R(w^0+x)\| & \\ \text{subject to} & e^T x & = & 0. \end{array}$$

The efficient frontier

$$\begin{array}{lll} \text{minimize} & r^T(w^0+x) - \lambda\|R(w^0+x)\| & \\ \text{subject to} & e^T x & = & 0. \end{array}$$

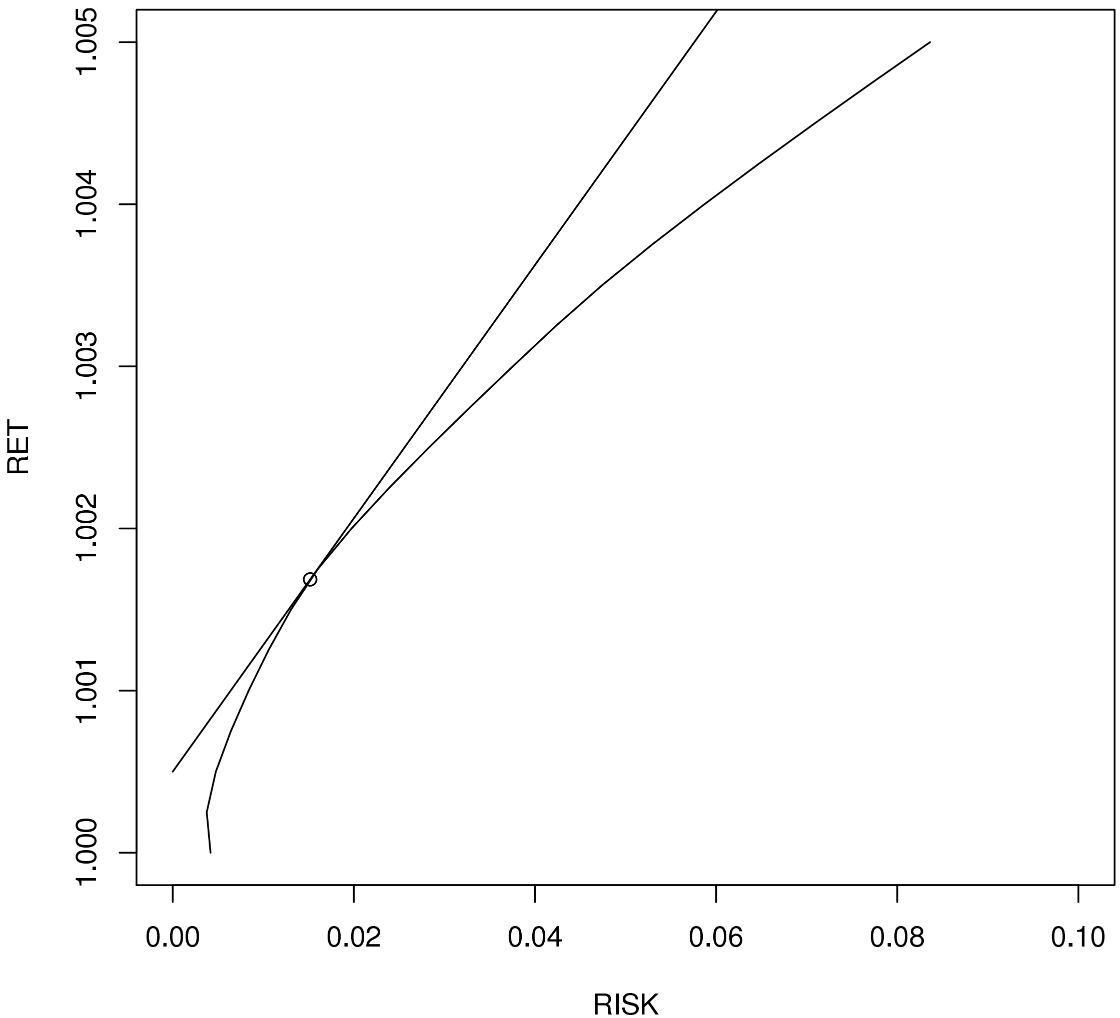
```
plot(RISK, RET, 'l',
     xlim = c(0, 0.1),
     ylim = c(tlow, thigh))

points(SHARPE_RISK, SHARPE_RET)

lines(x = c(0,1),
      y = rf*sum(w0) + SHARPE_RATIO*c(0,1))
```

The efficient frontier

$$\begin{array}{lll} \text{minimize} & r^T(w^0 + x) - \lambda \|R(w^0 + x)\| & \\ \text{subject to} & e^T x & = \quad 0. \end{array}$$



High performance

Column generation

Multiple knapsack problem:

- One pool of (weight, value)-items
- N knapsacks of limited capacity
- Maximize profit

Column generation

Multiple knapsack problem:

- One pool of (weight, value)-items
- N knapsacks of limited capacity
- Maximize profit

One subproblem per knapsack (MILP):

- Select items for the individual knapsack
- Maximize reduced-cost-profit

One master (LP):

- Select item-disjoint solutions for each knapsack
- Maximize profit

Solving subproblems

TASK: Update and solve subproblems in parallel.

```
>> require("doMC")
Loading required package:  doMC
Loading required package:  foreach
Loading required package:  iterators
Loading required package:  codetools
Loading required package:  multicore

>> registerDoMC(cores = 16)
```

Solving subproblems


```
mosek_clean()

SPsols <-
foreach(i = 1:nsack) %dopar% {
   prob <- updateObj(SP[[i]],
                     itemProfits,
                     itemDuals,
                     sackDuals[i],
                     MPfeasibility)
   sol <- getSolution(prob)

   if (isImproving(prob, sol)) {
     return(sol)
   } else {
     return(NULL)
   }
}
```

Performance

200 knapsacks of:

- capacity = $1, \ 2, \ ..., \ 200$

200 items of:

- weight = $1, \ 2, \ ..., \ 200$
- profit = $200, \ 199, \ ..., \ 1$

Performance

200 knapsacks of:

- capacity = 1, 2, ..., 200

200 items of:

- weight = 1, 2, ..., 200
- profit = 200, 199, ..., 1

Optimality in **449** seconds:

- **37600** subproblems solved (**17117** columns added)
- **187** master problems solved

Performance

200 knapsacks of:

- capacity = 1, 2, ..., 200

200 items of:

- weight = 1, 2, ..., 200
- profit = 200, 199, ..., 1

Optimality in **449** seconds:

- **37600** subproblems solved (**17117** columns added)
- **187** master problems solved

Average speed: **84** problems per second.

Summary

More information

The R project

- `r-project.org`

Home of MOSEK ApS

- `mosek.com`

Need help? MOSEK Google Group!

- `groups.google.com/group/mosek`

The Rmosek introduction page

- `cran.r-project.org/web/packages/Rmosek/index.html`

The Rmosek development site

- `rmosek.r-forge.r-project.org`

Conclusion

Goals:

- ✓ Open-source
- ✓ Highly effective
- ✓ Integrate with what R users normally do

Conclusion

Goals:

- ✓ Open-source
- ✓ Highly effective
- ✓ Integrate with what R users normally do

Thank you!