Comparison of two versions of turnover constraint optimization

Kirk Li

October 12, 2013

1 source scripts, load data

```
# global chunk options
library(knitr)
opts_chunk$set(cache = TRUE, tidy = FALSE, autodep = TRUE, fig.width = 6, fig.height = 6
```

```
inslib <- function(x){
x <-as.character(substitute(x))
if(!x %in% rownames(installed.packages()))
{install.packages(x)}
eval(parse(text=paste("library(",x,")",sep="")))}
inslib("quadprog")
inslib("xts")
inslib("corpcor")
inslib(knitr) # inslib works w/ "
source("turnoveroptdoug.r")
source("TurnoverOpt.R")
source("mvo.constrained.r")
source("efront.constrained.r")
source("barplot.wts.r")
load("crsp.short.Rdata")
returns = midcap.ts[,1:5]
returns = coredata(returns)
```

2 scenario 1: without mean constraints, with shorting allowed:

```
mu.target = NULL
long.only = FALSE
toc=turnover=0.2
wts.initial=w.initial=rep(1/ncol(returns),ncol(returns))
res1 <- TurnoverOpt_doug(returns, mu.target =mu.target,
wts.initial = wts.initial, toc = toc,long.only=long.only)
res2 <- TurnoverOpt(returns,mu.target=mu.target,
w.initial=wts.initial,turnover=turnover,long.only=long.only)
RES <- rbind(c(res1$wts,res1$port.mu,res1$port.var,res1$turnover),
c(res2$w,res2$port.mu, res2$port.var, res2$achieved.turnover))
colnames(RES) <- c(rep("wt",5),"mu","var","turnover")
rownames(RES) <- c("doug","previous")
RES

##            wt     wt     wt     wt  wt      mu      var turnover
## doug     0.2 0.1640 0.1870 0.1490 0.3 0.01100 0.004000      0.2
## previous 0.2 0.1639 0.1875 0.1486 0.3 0.01126 0.004163      0.2
```

3 scenario 2: without mean constraints, without shorting:

```
mu.target = NULL
long.only = TRUE
toc=turnover=0.2
wts.initial=w.initial=rep(1/ncol(returns),ncol(returns))
res1 <- TurnoverOpt_doug(returns, mu.target =mu.target,
wts.initial = wts.initial, toc = toc,long.only=long.only)
res2 <- TurnoverOpt(returns,mu.target=mu.target,
w.initial=wts.initial,turnover=turnover,long.only=long.only)
RES <- rbind(c(res1$wts,res1$port.mu,res1$port.var,res1$turnover),
c(res2$w,res2$port.mu, res2$port.var, res2$achieved.turnover))
colnames(RES) <- c(rep("wt",5),"mu","var","turnover")
rownames(RES) <- c("doug","previous")
RES

##            wt     wt     wt     wt  wt      mu      var turnover
## doug     0.2 0.1640 0.1870 0.1490 0.3 0.01100 0.004000      0.2
## previous 0.2 0.1639 0.1875 0.1486 0.3 0.01126 0.004163      0.2
```

4 scenario 3: with mean constraints, with shorting allowed:

```
mu.target = 0.01
long.only = FALSE
toc=turnover=0.5
res1 <- TurnoverOpt_doug(returns, mu.target =mu.target,
wts.initial = wts.initial, toc = toc, long.only=long.only)
res2 <- TurnoverOpt(returns,mu.target=mu.target,
w.initial=wts.initial,turnover=turnover,long.only=long.only)

RES <- rbind(c(res1$wts,res1$port.mu,res1$port.var,res1$turnover),
c(res2$w,res2$port.mu, res2$port.var, res2$achieved.turnover))
colnames(RES) <- c(rep("wt",5),"mu","var","turnover")
rownames(RES) <- c("doug","previous")
RES

##              wt     wt      wt     wt     wt   mu      var turnover
## doug     0.2370 0.1780 0.06900 0.1020 0.4130 0.01 0.004000      0.5
## previous 0.2366 0.1782 0.06946 0.1023 0.4134 0.01 0.003828      0.5
```

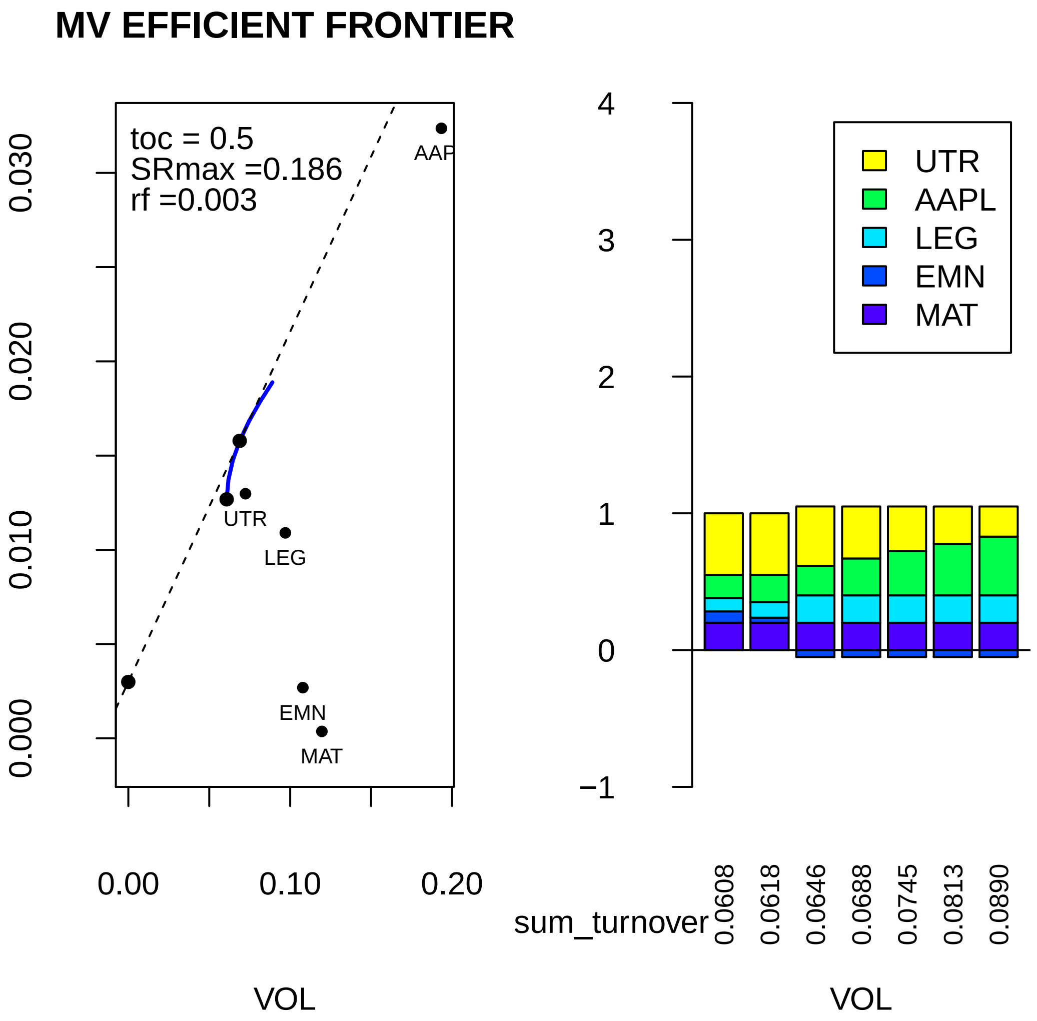
5 scenario 4: with mean constraints, without shorting:

```
mu.target = 0.01
long.only = TRUE
toc=turnover=0.5
res1 <- TurnoverOpt_doug(returns, mu.target =mu.target,
wts.initial = wts.initial, toc = toc, long.only=long.only)
res2 <- TurnoverOpt(returns,mu.target=mu.target,
w.initial=wts.initial,turnover=turnover,long.only=long.only)

RES <- rbind(c(res1$wts,res1$port.mu,res1$port.var,res1$turnover),
c(res2$w,res2$port.mu, res2$port.var, res2$achieved.turnover))
colnames(RES) <- c(rep("wt",5),"mu","var","turnover")
rownames(RES) <- c("doug","previous")
RES

##              wt     wt      wt     wt     wt   mu      var turnover
## doug     0.2370 0.1780 0.06900 0.1020 0.4130 0.01 0.004000      0.5
## previous 0.2366 0.1782 0.06946 0.1023 0.4134 0.01 0.003828      0.5
```

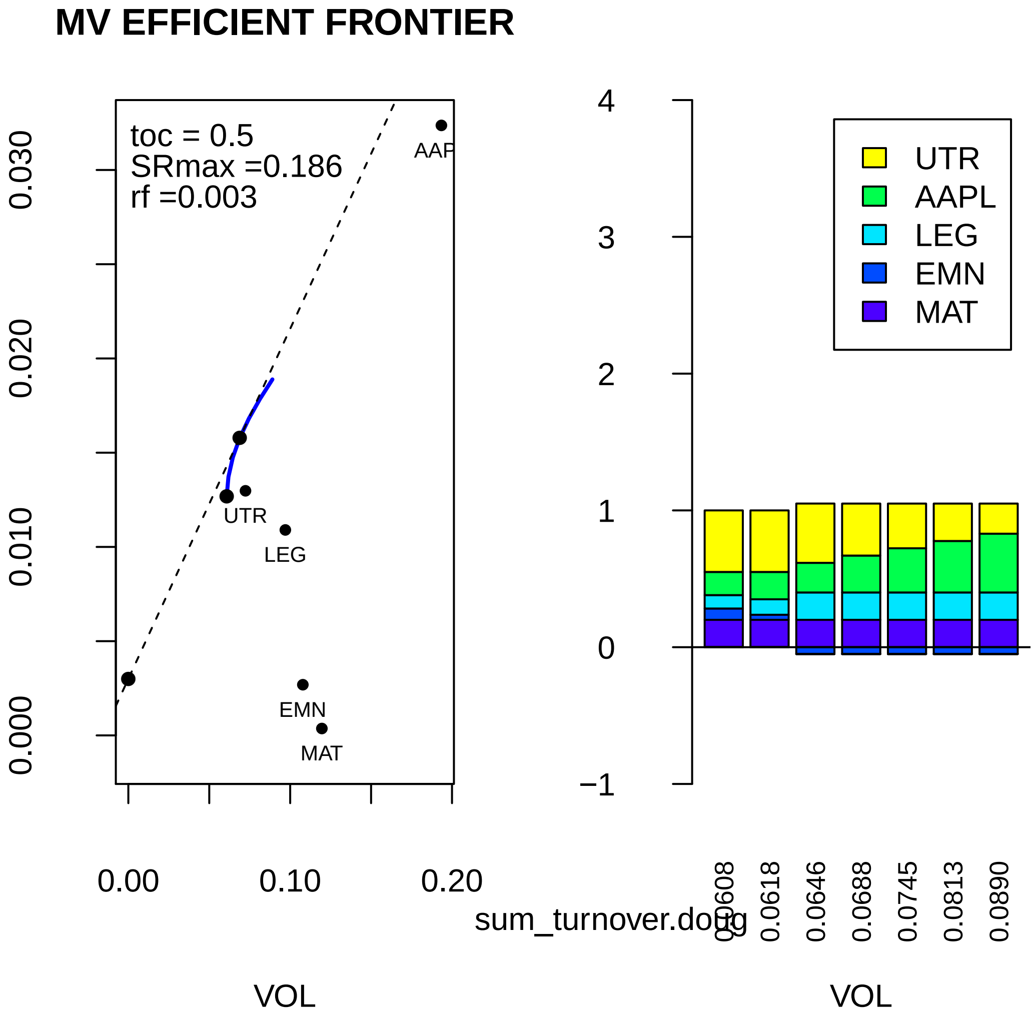
6 efficient frontier plot:

Here we compare two versions of turnover on efficient frontier plot, using two different turnover (0.5, 10). We found there is no difference in two versions.

```
sum=1 # full investment
mu.target=NULL
w.initial=rep(1/n.stocks,n.stocks)
toc=0.5
digits=4
wts.only=T
mu.min = NULL
mu.max = NULL
rf = .003
npoints = 20
wts.plot = T
printout = F
bar.ylim = c(-1,4)
clist <- c("sum","turnover")
list.arg <- list(
sum=sum,
toc=toc,
w.initial=w.initial)
cset <- NULL
cset <-combine.cset(clist=clist,returns=returns,list.arg)

## sum
## turnover

efrontPlot(returns, cset, rf = .003, npoints = 20,wts.plot = T,
bar.ylim = c(-1,4),list.arg=list.arg)

## [1] "turnover/propcost constraints reduced the max mean return in efficient frontier

mtext(paste(clist,collapse="_"),side=1,line=3)
```

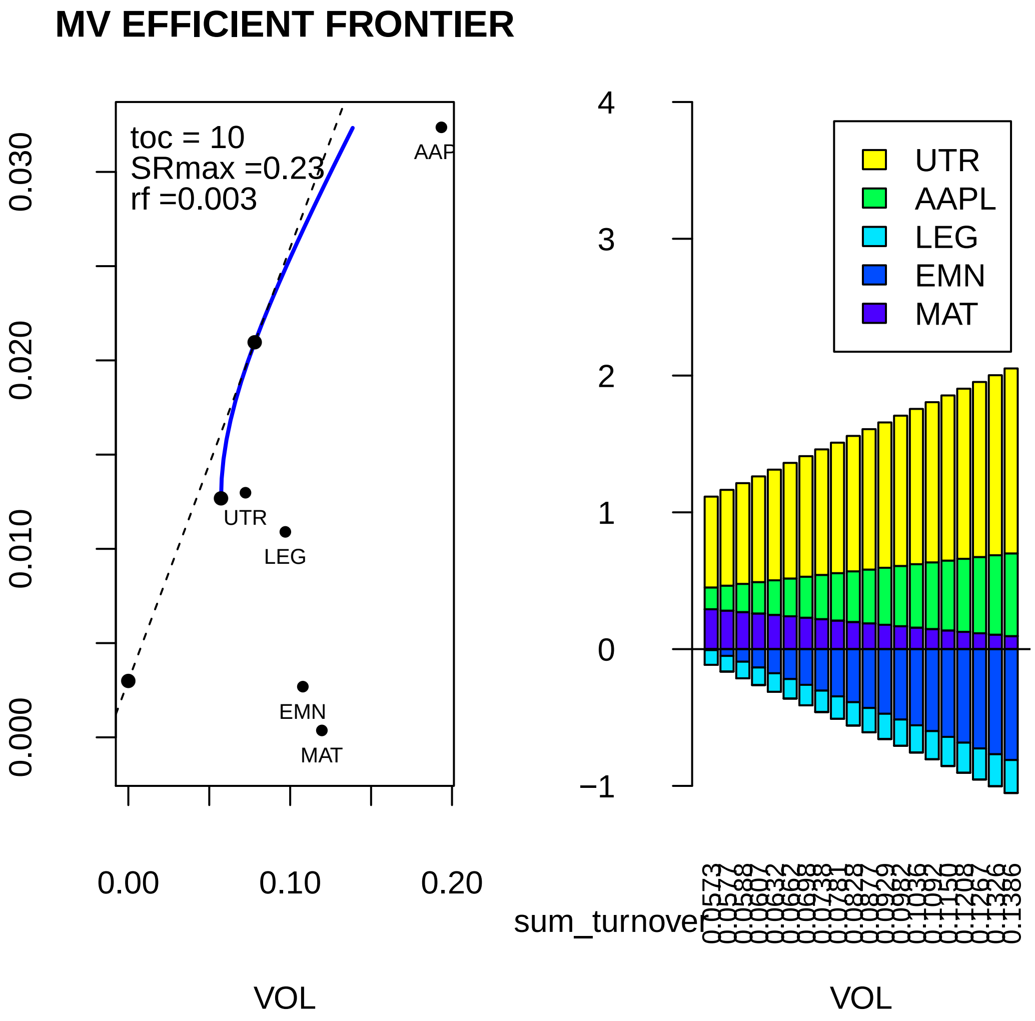


```
clist <- c("sum","turnover.doug")
list.arg <- list(
sum=sum,
toc=toc,
w.initial=w.initial)
cset <- NULL
cset <-combine.cset(clist=clist,returns=returns,list.arg)

## sum
## turnover.doug


efrontPlot(returns, cset, rf = .003, npoints = 20,wts.plot = T,
bar.ylim = c(-1,4),list.arg=list.arg)

## [1] "turnover/propcost constraints reduced the max mean return in efficient frontier

mtext(paste(clist,collapse="_"),side=1,line=3)
```

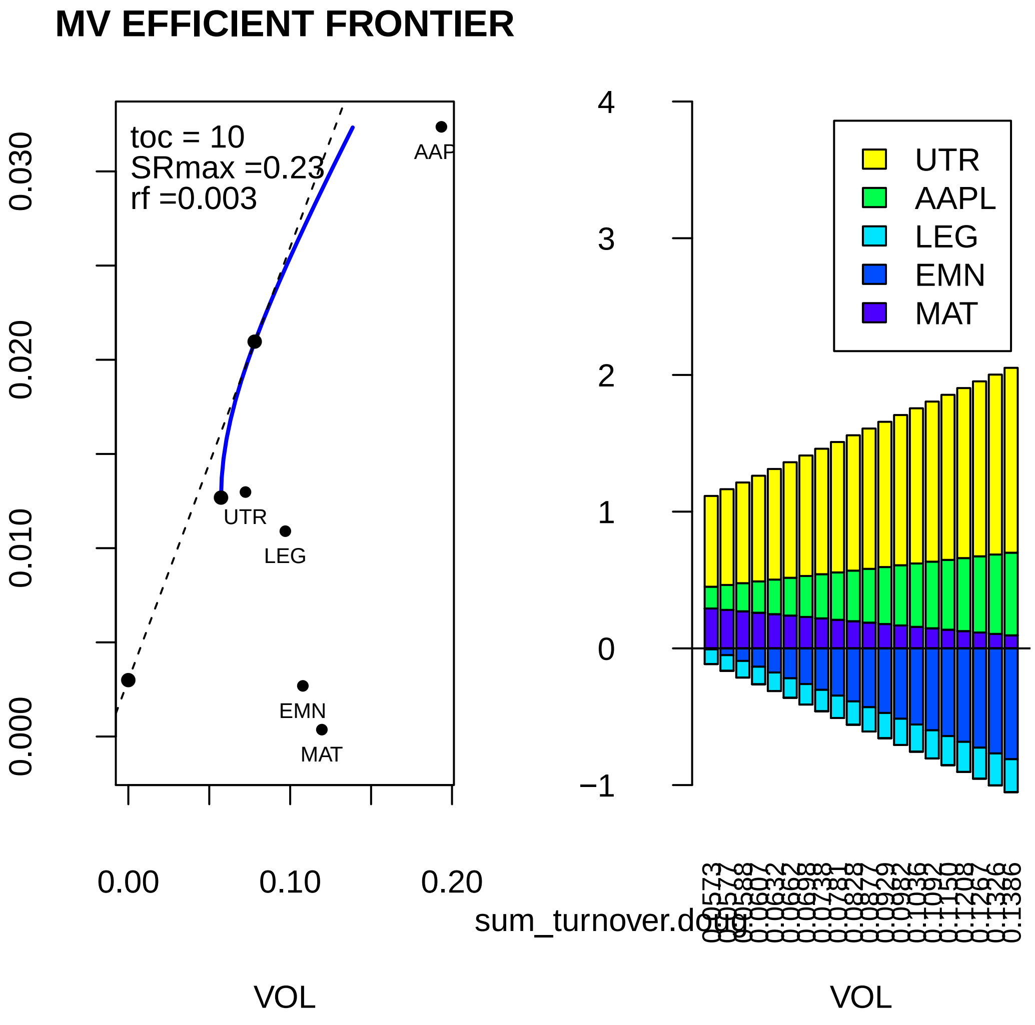


Now we assign a large number toc:

```
toc=10
clist <- c("sum","turnover")
list.arg <- list(
sum=sum,
toc=toc,
w.initial=w.initial)
cset <- NULL
cset <-combine.cset(clist=clist,returns=returns,list.arg)

## sum
## turnover

efrontPlot(returns, cset, rf = .003, npoints = 20,wts.plot = T,
bar.ylim = c(-1,4),list.arg=list.arg)
mtext(paste(clist,collapse="_"),side=1,line=3)
```



```
clist <- c("sum","turnover.doug")
list.arg <- list(
sum=sum,
toc=toc,
w.initial=w.initial)
cset <- NULL
cset <-combine.cset(clist=clist,returns=returns,list.arg)

## sum
## turnover.doug


efrontPlot(returns, cset, rf = .003, npoints = 20,wts.plot = T,
bar.ylim = c(-1,4),list.arg=list.arg)
mtext(paste(clist,collapse="_"),side=1,line=3)
```


MV EFFICIENT FRONTIER
toc = 10
SRmax =0.23
rf =0.003
AAPL
UTR
LEG
EMN
MAT
VOL
sum_turnover.doug
UTR
AAPL
LEG
EMN
MAT
VOL