Strategies: borrowing ideas from Engineering Returns

December 19, 2011

By systematicinvestor

(This article was first published on **Systematic Investor » R**, and kindly contributed to R-bloggers)

Frank Hassler at Engineering Returns blog wrote an excellent article Rotational Trading: how to reduce trades and improve returns. The article presents four methods to reduce trades:

- Trade less frequently. I.e. weekly instead of daily rebalancing.
- Different criteria for enter / exit a trade.
- Smooth the rank over the last couple of bars.
- Combination of above.

I want show how to implement these ideas using the backtesting library in theSystematic Investor Toolbox. I will use the 21 ETFs from the ETF Sector Strategypost as the investment universe.

Following code loads historical prices from Yahoo Fiance and compares performance of the daily versus weekly rebalancing using the backtesting library in the Systematic Investor Toolbox:

```
# Load Systematic Investor Toolbox (SIT)
setInternet2(TRUE)
con = gzcon(url('https://github.com/systematicinves
    source(con)
close(con)

    #*****************************************
    # Load historical data
    #*****************************************
    load.packages('quantmod')
    tickers = spl('XLY,XLP,XLE,XLF,XLV,XLI,XLB,XLK,

    data <- new.env()
    getSymbols(tickers, src = 'yahoo', from = '1970
        for(i in ls(data)) data[[i]] = adjustOHLC(
    bt.prep(data, align='remove.na', dates='1970::2

    #*****************************************
    # Code Strategies : weekly rebalancing
    #*****************************************
    prices = data$prices
    n = len(tickers)

    # find week ends
    week.ends = endpoints(prices, 'weeks')
        week.ends = week.ends[week.ends > 0]


    # Rank on ROC 200
    position.score = prices / mlag(prices, 200)
        position.score.ma = position.score
        buy.rule = T
```

```
33
34      # Select Top 2 funds daily
35      data$weight[] = NA
36          data$weight[] = ntop(position.score, 2)
37          capital = 100000
38          data$weight[] = (capital / prices) * bt.exr
39      top2.d = bt.run(data, type='share', trade.summa
40
41      # Select Top 2 funds weekly
42      data$weight[] = NA
43          data$weight[week.ends,] = ntop(position.sc
44          capital = 100000
45          data$weight[] = (capital / prices) * bt.exr
46      top2.w = bt.run(data, type='share', trade.summa
47
48      # Plot Strategy Metrics Side by Side
49      plotbt.strategy.sidebyside(top2.d, top2.w, per
```

System	top2.d	top2.w
Period	Aug2001 - Dec2011	Aug2001 - Dec2011
Cagr	2.4	4.6
DVR	3	9.2
Sharpe	22	31.8
R2	13.8	28.9
Win.Percent	52.8	55.5
Avg.Trade	0.1	0.4
MaxDD	-55.8	-54.7
Num.Trades	443	164

The number of trades falls down from 443 to 164 as we switch from daily to weekly rebalancing. The additional bonus is the better returns for the weekly rebalancing.

Next, let's examine different entry/exit rank. We will buy top 2 ETFs and will keep them till their ranks drop below 4 /6.

```
1   #*****************************************************
2   # Code Strategies : different entry/exit rank
3   #*****************************************************
4
5   # Select Top 2 funds, Keep till they are in 4/6 ran
6   data$weight[] = NA
7       data$weight[] = ntop.keep(position.score, 2, 4)
8       capital = 100000
```

```
    data$weight[] = (capital / prices) * bt.exrem(c
top2.d.keep4 = bt.run(data, type='share', trade.sur

data$weight[] = NA
    data$weight[] = ntop.keep(position.score, 2, 6)
    capital = 100000
    data$weight[] = (capital / prices) * bt.exrem(c
top2.d.keep6 = bt.run(data, type='share', trade.sur

# Plot Strategy Metrics Side by Side
plotbt.strategy.sidebyside(top2.d, top2.d.keep4, tc
```

System	top2.d	top2.d.keep4	top2.d.keep6
Period	Aug2001 - Dec2011	Aug2001 - Dec2011	Aug2001 - Dec2011
Cagr	2.4	5.4	5.3
DVR	3	17.9	11.8
Sharpe	22	35.5	34.3
R2	13.8	50.4	34.5
Win.Percent	52.8	60	55.8
Avg.Trade	0.1	0.8	1.6
MaxDD	-55.8	-47.6	-54.1
Num.Trades	443	95	52

The number of trades falls down from 443 to 95 to 52 as we hold on to our selection for longer periods.

Next, let's examine rank smoothing. Instead of using the most recent rank, we will use different averages of rank's recent values.

```
#*****************************************************
# Code Strategies : Rank smoothing
#*****************************************************

models = list()
models$Bench = top2.d
for( avg in spl('SMA,EMA') ) {
    for( i in c(3,5,10,20) ) {
        position.score.smooth = bt.apply.matrix(pos
            position.score.smooth[!buy.rule,] = NA

        data$weight[] = NA
            data$weight[] = ntop(position.score.smo
            capital = 100000
            data$weight[] = (capital / prices) * bt
```

```
        models[[ paste(avg,i) ]] = bt.run(data, typ
    }
}

# Plot Strategy Metrics Side by Side
plotbt.strategy.sidebyside(models, perfromance.fn =
```

System	Bench	SMA 3	SMA 5	SMA 10	SMA 20	EMA 3	EMA 5	EMA 10	EMA 20
Period	Aug2001 - Dec2011	Aug2001 - Dec2011	Aug2001 - Dec2011	Aug2001 - Dec2011	Aug2001 - Dec2011	Aug2001 - Dec2011	Aug2001 - Dec2011	Aug2001 - Dec2011	Aug2001 - Dec2011
Cagr	2.4	2.9	4.4	5.4	6.5	2.3	3.3	4.6	5.3
DVR	3	4.1	8	12.9	19.1	2.7	4.3	9.1	11.5
Sharpe	22	24.2	30.5	34.8	39.4	21.5	26	31.4	34.2
R2	13.8	16.8	26.3	37.1	48.4	12.8	16.5	29	33.6
Win.Percent	52.8	52.4	54.4	57.5	59.2	50.6	53.5	55.4	59.3
Avg.Trade	0.1	0.2	0.4	0.6	0.9	0.2	0.3	0.6	0.9
MaxDD	-55.8	-54.1	-54.6	-51.8	-49	-56.3	-55.7	-54.2	-50.9
Num.Trades	443	248	180	134	98	247	200	130	91

The number of trades falls down as we increase the length of period used in averaging. There is no big difference in using simple moving average (SMA) versus exponential smoothing average (EMA).

Next, let's combine different methods to reduce number of trades.

```
#*****************************************************
# Code Strategies : Combination
#*****************************************************

# Select Top 2 funds daily, Keep till they are 6 ra
position.score.smooth = bt.apply.matrix(position.sc
    position.score.smooth[!buy.rule,] = NA
data$weight[] = NA
    data$weight[] = ntop.keep(position.score.smooth
    capital = 100000
    data$weight[] = (capital / prices) * bt.exrem(c
top2.d.keep6.EMA10 = bt.run(data, type='share', tra

# Select Top 2 funds weekly, Keep till they are 6 r
data$weight[] = NA
    data$weight[week.ends,] = ntop.keep(position.sc
    capital = 100000
    data$weight[] = (capital / prices) * bt.exrem(c
top2.w.keep6 = bt.run(data, type='share', trade.sum

# Select Top 2 funds weekly, Keep till they are 6 r
position.score.smooth[] = NA
    position.score.smooth[week.ends,] = bt.apply.ma
        position.score.smooth[!buy.rule,] = NA

data$weight[] = NA
    data$weight[week.ends,] = ntop.keep(position.sc
    capital = 100000
    data$weight[] = (capital / prices) * bt.exrem(c
top2.w.keep6.EMA10 = bt.run(data, type='share', tra
```

```
31
32  # Plot Strategy Metrics Side by Side
33  plotbt.strategy.sidebyside(top2.d, top2.d.keep6, tc
```

System	top2.d	top2.d.keep6	top2.d.keep6.EMA10	top2.w	top2.w.keep6	top2.w.keep6.EMA10
Period	Aug2001 - Dec2011	Aug2001 - Dec2011	Aug2001 - Dec2011	Aug2001 - Dec2011	Aug2001 - Dec2011	Aug2001 - Dec2011
Cagr	2.4	5.3	3.9	4.6	6.7	7.3
DVR	3	11.8	10.1	9.2	22	15.3
Sharpe	22	34.3	28.3	31.8	39.4	41
R2	13.8	34.5	35.6	28.9	55.8	37.2
Win.Percent	52.8	55.8	65.8	55.5	61.9	71.4
Avg.Trade	0.1	1.6	1.8	0.4	2.1	3.8
MaxDD	-55.8	-54.1	-56.5	-54.7	-54.2	-58.9
Num.Trades	443	52	38	164	42	28

The overall winner is a weekly strategy that buys top 2 ETF's based on 10 week exponential average rank and keeps them till their ranks drop below 6. The number of trades falls down from 443 to 28 and performance (CAGR) goes up from 2.4% to 7.3%.

The next step, which you can do as a homework, is to find ways to control the strategy's drawdowns. One solution is discussed in the Avoiding severe draw downs post.

To view the complete source code for this example, please have a look at thebt.rotational.trading.trades.test() function in bt.test.r at github.