



Not Over Thinking

| Asset Class Trend Following

Algorithmic Trading Strategy with Full Code

Haixiang

2023.03 | Vol 12.

Hxyan.2015@gmail.com | github.com/hxyan2020

STRATEGY & ECONOMIC RATIONALE

The investment universe consists of 5 ETFs (SPY – US stocks, EFA – foreign stocks, BND – bonds, VNQ – REITs, GSG – commodities), and portfolio is equally weighted.

LONG	SHORT
Hold each asset class ETF only when it is over its ten-month Simple Moving Average, otherwise stay in cash.	When it drops below ten-month SMA

PARAMETER & VARIABLES

PARAMETER	VALUE
MARKETS TRADED	bonds, commodities, equities, REITs
FINANCIAL INSTRUMENTS	CFDs, ETFs, funds, futures
REGION	Global
PERIOD OF REBALANCING	Monthly
NO. OF TRADED INSTRUMENTS	5 instruments + cash
WEIGHTING	Equal weighting
LOOKBACK PERIODS	10 months (210 days)
HOLDING PERIODS	Depends
LONG/SHORT	Long Only

ALGORITHM

```
class AssetClassTrendFollowing(QCAlgorithm):

    def Initialize(self):
        self.SetStartDate(2000, 1, 1)
        self.SetCash(100000)

        self.sma = {}
        period = 10 * 21
        ## 10 months * 21 trading days per month
        self.SetWarmUp(period, Resolution.Daily) ##look-back period

        self.symbols = ["SPY", "EFA", "IEF", "VNQ", "GSG"]
        self.rebalance_flag = False

        self.tracked_symbol = None
        for symbol in self.symbols:
            self.AddEquity(symbol, Resolution.Minute)
            self.sma[symbol] = self.SMA(symbol, period, Resolution.Daily)
            ## use 210 days to warm up SMA for each of the symbols

        self.recent_month = -1
```

```
def OnData(self, data):
    if self.IsWarmingUp: return

    if not (self.Time.hour == 9 and self.Time.minute == 31):
        return ## start to trade from 09:31 daily

    # rebalance once a month
    if self.Time.month == self.recent_month:
        return
    self.recent_month = self.Time.month ## update recent_month when new month arrives

    long = [ symbol for symbol in self.symbols if symbol in data and data[symbol] and
self.sma[symbol].IsReady and data[symbol].Value > self.sma[symbol].Current.Value ]

    # trade execution
    invested = [x.Key.Value for x in self.Portfolio if x.Value.Invested]
    for symbol in invested:
        if symbol not in long:
            self.Liquidate(symbol)

    for symbol in long:
        self.SetHoldings(symbol, 1 / len(long))
```

BACKTESTING PERFORMANCE



Fig 1. Overall Performance

PSR	0.111%	Sharpe Ratio	0.475
Total Trades	823	Average Win	0.66%
Average Loss	-1.33%	Compounding Annual Return	6.458%
Drawdown	29.300%	Expectancy	0.263
Net Profit	324.510%	Loss Rate	16%
Win Rate	84%	Profit-Loss Ratio	0.50
Alpha	0.033	Beta	0.297
Annual Standard Deviation	0.105	Annual Variance	0.011
Information Ratio	-0.053	Tracking Error	0.147
Treynor Ratio	0.168	Total Fees	\$2654.40
Estimated Strategy Capacity	\$3200000.00	Lowest Capacity Asset	EFA S79U6IHK5HLX

Fig 2. Performance Metrics

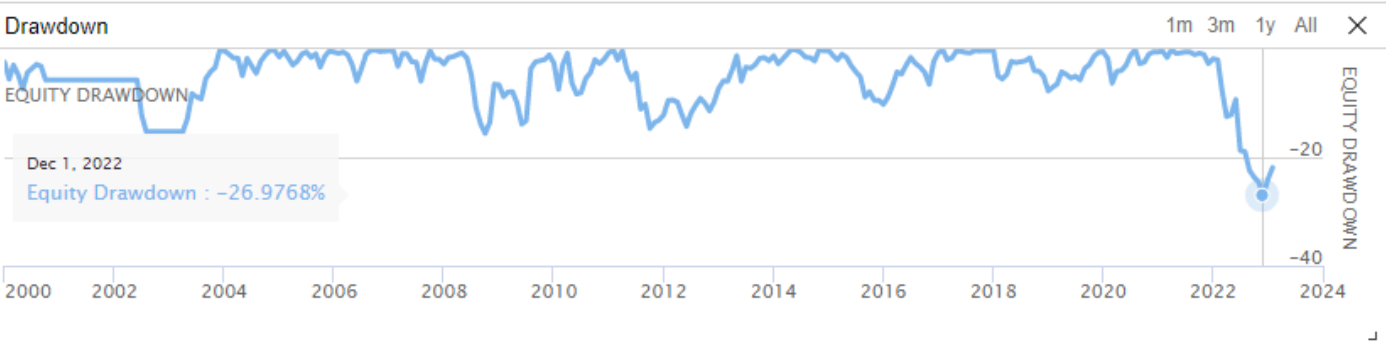


Fig 3. Drawdown



Fig 4. Assets Sales Volume