Not Over Thinking

Short Interest Effect - Long Only

Algorithmic Trading Strategy with Full Code

Haixiang

2024.02 | Vol 76.

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

STRATEGY & ECONOMIC RATIONALE

All stocks from NYSE, AMEX, and NASDAQ are part of the investment universe. The short-interest ratio is used as the predictor variable. Stocks are sorted based on their short interest ratio, and the first percentile is held. The portfolio is equally weighted and rebalanced monthly.

BUY	SELL
(see above)	(see above)

PARAMETER & VARIABLES

PARAMETER	VALUE
MARKETS TRADED	Equity
FINANCIAL INSTRUMENTS	Stocks
REGION	United States
PERIOD OF REBALANCING	Monthly
NO. OF TRADED INSTRUMENTS	230
WEIGHTING	Equal weighting
LOOKBACK PERIODS	N/A
LONG/SHORT	Long only

ALGORITHM

```
from AlgorithmImports import *
class ShortInterestEffect(QCAlgorithm):

    def Initialize(self):
        self.SetStartDate(2010, 1, 1)
        self.SetCash(100000)

        # NOTE: We use only s&p 100 stocks so it's possible to fetch short interest data from q
uandl.
        self.symbols = [
            'AAPL','MSFT','AMZN','FB','BRK.B','GOOGL','GOOG','JPM','JNJ','V','PG','XOM','UNH','
BAC','MA','T','DIS','INTC','HD','VZ','MRK','PFE',
            'CVX','KO','CMCSA','CSCO','PEP','WFC','C','BA','ADBE','WMT','CRM','MCD','MDT','BMY',
'ABT','NVDA','NFLX','AMGN','PM','PYPL','TMO',
            'COST','ABBV','ACN','HON','NKE','UNP','UTX','NEE','IBM','TXN','AVGO','LLY','ORCL','
LIN','SBUX','AMT','LMT','GE','MMM','DHR','QCOM',
            'CVS','MO','LOW','FIS','AXP','BKNG','UPS','GILD','CHTR','CAT','MDLZ','GS','USB','CI
','ANTM','BDX','TJX','ADP','TFC','CME','SPGI',
            'COP','INTU','ISRG','CB','SO','D','FISV','PNC','DUK','SYK','ZTS','MS','RTN','AGN','
BLK'
            ]

        for symbol in self.symbols:
            data = self.AddEquity(symbol, Resolution.Daily)
            data.SetFeeModel(CustomFeeModel())
            data.SetLeverage(5)

            self.AddData(QuandlFINRA_ShortVolume, 'FINRA/FNSQ_' + symbol, Resolution.Daily)

        self.recent_month = -1

    def OnData(self, data):
        if self.recent_month == self.Time.month:
            return
        self.recent_month = self.Time.month

        short_interest = {}
        for symbol in self.symbols:
            sym = 'FINRA/FNSQ_' + symbol
            if sym in data and data[sym] and symbol in data and data[symbol]:
                short_vol = data[sym].GetProperty("SHORTVOLUME")
                total_vol = data[sym].GetProperty("TOTALVOLUME")

                short_interest[symbol] = short_vol / total_vol

        long = []
        if len(short_interest) >= 10:
            sorted_by_short_interest = sorted(short_interest.items(), key = lambda x: x[1], rev
erse = True)
            decile = int(len(sorted_by_short_interest) / 10)
            long = [x[0] for x in sorted_by_short_interest[-decile:]]

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

        for symbol in long:
            self.SetHoldings(symbol, 1 / len(long))
            class QuandlFINRA_ShortVolume(PythonQuandl):
    def __init__(self):
        self.ValueColumnName = 'SHORTVOLUME'    # also 'TOTALVOLUME' is accesible.
# Custom fee model.class CustomFeeModel(FeeModel):
    def GetOrderFee(self, parameters):
        fee = parameters.Security.Price * parameters.Order.AbsoluteQuantity * 0.00005
        return OrderFee(CashAmount(fee, "USD"))
```

BACKTESTING PERFORMANCE

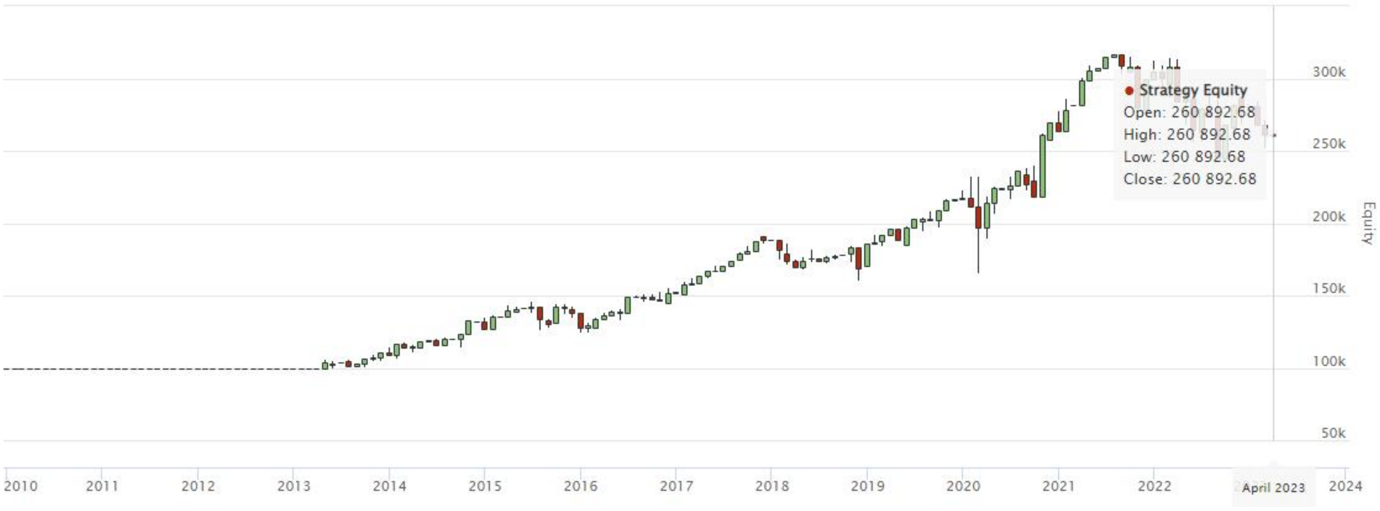
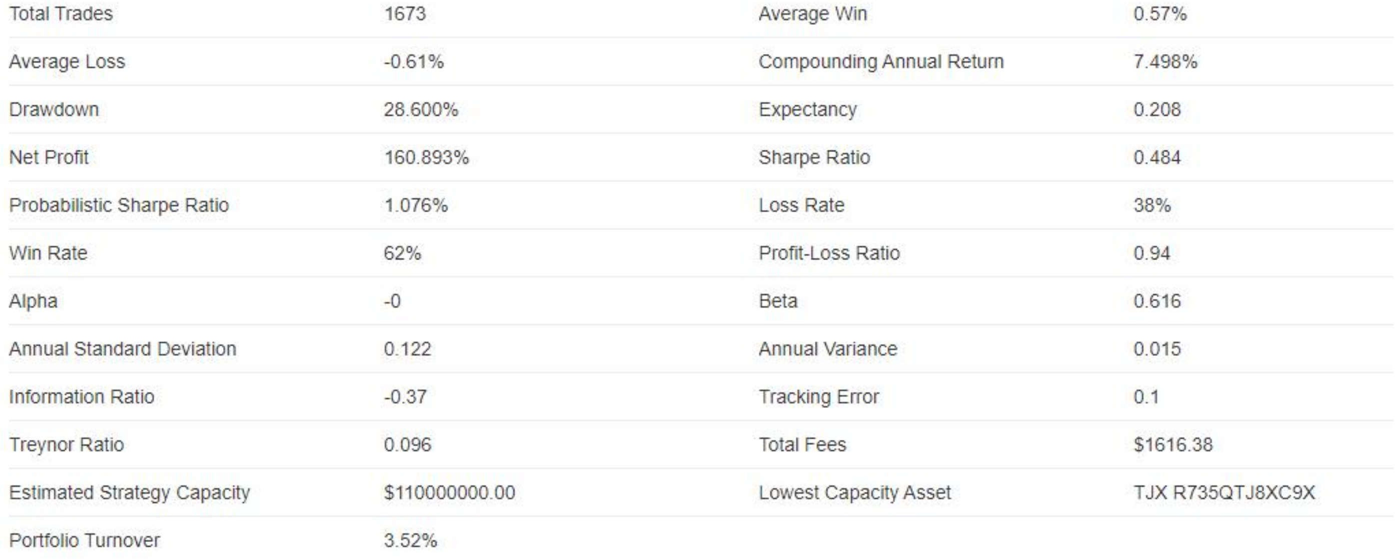


Fig 1. Overall Performance



Total Trades	1673	Average Win	0.57%
Average Loss	-0.61%	Compounding Annual Return	7.498%
Drawdown	28.600%	Expectancy	0.208
Net Profit	160.893%	Sharpe Ratio	0.484
Probabilistic Sharpe Ratio	1.076%	Loss Rate	38%
Win Rate	62%	Profit-Loss Ratio	0.94
Alpha	-0	Beta	0.616
Annual Standard Deviation	0.122	Annual Variance	0.015
Information Ratio	-0.37	Tracking Error	0.1
Treynor Ratio	0.096	Total Fees	$1616.38
Estimated Strategy Capacity	$110000000.00	Lowest Capacity Asset	TJX R735QTJ8XC9X
Portfolio Turnover	3.52%		

Fig 2. Performance Metrics