Abstract

I do research on the paper A Game-Theoretical Approach for Designing Market Trading Strategies wrote by Garrison W. Greenwood and Richard Tymerski, and replicate some of the interesting ideas and methods in this paper.

This paper uses co-evolve fuzzy trading rules from market trend features which often appear only in partial form to design a strategy for trading shares of a single stock. Meanwhile, the paper shows the similarity between game theory and trading strategy. Stock market investment is naturally expressed as a game. Good trading strategies attract more investor dollars while poor strategies discourage further investment. Therefore, if we can choose a score to demonstrate the fitness of one strategy, and use this score to select good strategies, we can finally obtain the best trading strategy by creating several offspring strategies and simulating evolutionary process like what is happening in natural world. This co-evolutionary method is based on the assumption that good genes can be passed through generations and good combinations of good genes can make the child stand out among peers. In the stock market world, it means good market trend features can be found out through trying different strategies and different combinations of features.

I replicate the technical features and the fuzzy member functions in the paper, and I change a little bit of the details of the main evolutionary algorithm. Finally, I use the model to predict for up-trending trading days and back test the result by comparing the cumulative returns of the strategy and S&P 500 index.

1. Introduction

As the rising number of investors and brokerage firms in the market, trading strategies are of all kind and they are proprietary which are not disclosed to outsiders. Strategies are keep changing because of the volatile nature of the market, so the strategy which can make more profit will continuously attract investor dollars and on the contrary, the bad strategies will lose investors. Therefore, everything in the market is just like a zero-sum game and what we are going to do is to construct a trading strategy to figure out conditions in market historical data that predicts subsequent up-trend or down-trend days.

2. Features and Fuzzy Membership Functions

2.1 Features

We want to find attributes from a company's past trading activities and uses technical indicators based on open, close, high and low prices of the stock to be the input features.

First of all, the NRk. We first compute the range of each trading days which equals to the high price minus the low price, then NRk is defined as whether today's range is less than the minimum range of the past (k-1) trading days. If yes, the indicator is 1, otherwise 0. NRk days represent volatility contraction. The greater the number of narrow range days, the greater the counter reaction in wide ranging days.

Secondly, the DOJI. DOJI indicates that the open and close for the trading day are within some small percentage (x) of each other. A DOJI means the market reflects temporary price indecision and often signals a reversal.

Define:

DOJI(x) = 1, when |O - C| <= x * (H - L); 0, otherwise.

where rho in [0.05,0.3).

Thirdly, the Hookday. A hook day occurs when the price opens outside the previous day's range and then proceeds to reverse direction, generally indicating a reaction to temporarily overbought or oversold market conditions.

Up hook day:

$$O[-1] < L[0] - delta$$

Down hook day:

$$O[-1] > H[0] + delta$$

where delta is between [0,10].

2.2 Fuzzy Membership Functions

For crisp rules, it must give either a ‘yes’ or a ‘no’ answer, and it cannot handle the situations that one feature is almost true. However, fuzzy answers can make that because it can provide fuzzy answers somewhere in between a ‘yes’ or ‘no’ by transforming a 'yes' or 'no' question into fuzzy answers. It maps days(D) onto the unit interval via a function $\mu(D)$. It returns a value between 0 and 1 indicating to what degree features are present. Then, the fuzzy output variable is defuzzied to produce a crisp value on the unit interval, which the desirability of buying or selling stock on the next trading day.

The fuzzy membership functions of ea ch indicator are defined as follows:

For NRk:

$$\mu_k(x) = \begin{cases} 0, & x < v_{min} \\ c(x - v_{min}), & x \in [v_{min}, v_{max}) \\ 1, & x >= v_{max} \end{cases}$$

Parameter values:

k	c	v_{min}	v_{max}
4	1/2	2	4
6	1/3	3	6
7	1/3	4	7

For DOJI:

$$\mu(x) = \begin{cases} 1 - \frac{x}{p}, & x \in [0, p] \\ 0, & x \in otherwise \end{cases}$$

For Hookday:

$$\mu(x) = \begin{cases} 0, & x < 1/2 \\ 2(x + 0.5), & x \in [-1/2, 0) \\ 1, & x >= 0 \end{cases}$$

3. Evolutionary Algorithm

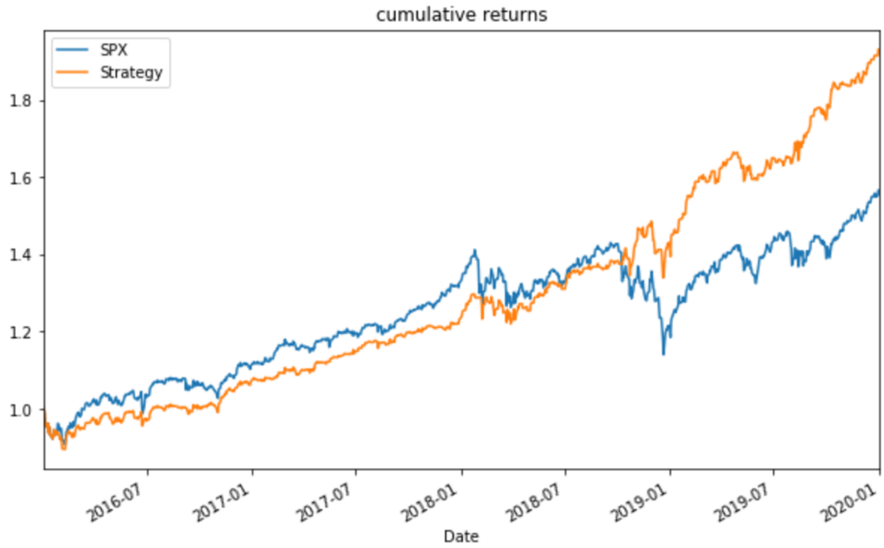
Evolutionary Algorithm is a generic population-based metaheuristic optimization algorithm. An EA uses mechanisms inspired by biological evolution, such as reproduction, mutation, recombination, and selection. Candidate solutions to the optimization problem play the role of individuals in a population, and the fitness function determines the quality of the solutions. Evolution of the population then takes place after the repeated application of the above operators.

After we calculate all the indicators and use their fuzzy membership functions to transform them respectively, we get all the original input data. Then, we can use evolutionary algorithm to find the best attributes which have the ability to tell the up-trend or down-trend days in the market.

In this part, I change a little bit of the method used in the paper and introduce my own strategy instead. The whole training dataset will be put into the model, and pipeline operators will modify the features. Then, it will combine the features and do recursive feature elimination. After that, the final classification is performed on the final dataset using random forest classifier.

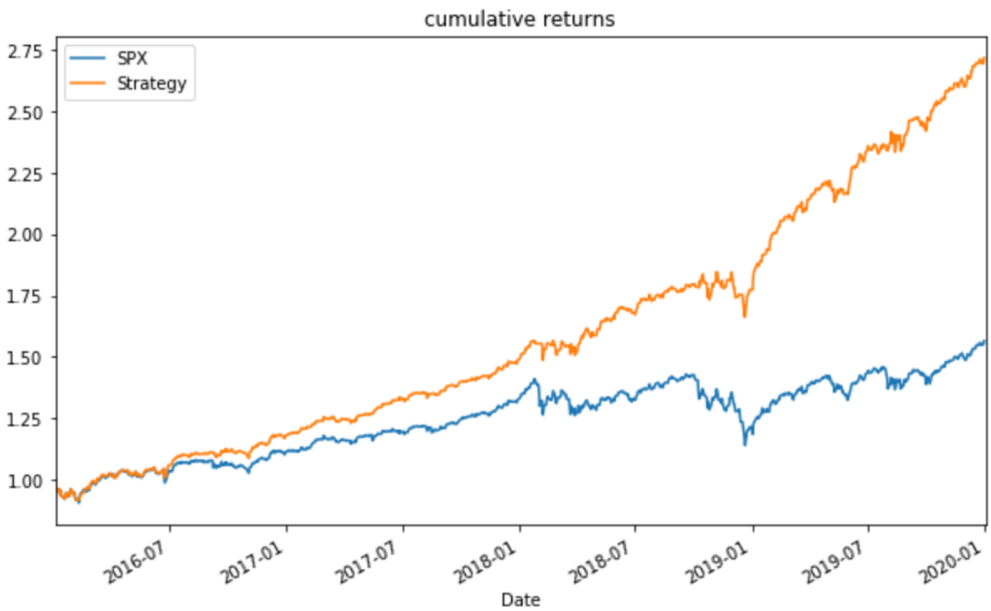
I use NR4, NR6, NR7, DOJI and Hookday as features and compute them using the formula and parameters aforementioned. The target equals 1 if today is up-trending day which means tomorrow's close price is higher than today's close price, and equals 0 if today is not up-trending day. The whole dataset is from 2010-01-01 to 2020-01-01 in total 10 years, and the training data is from 2010-01-01 to 2015-12-31, and testing data is from 2016-01-01 to 2020-01-01.

Firstly, I try evolutionary model in which the generation equals 100, population size equals 50, mutation rate equals 0.9, crossover rate equals 0.1, level of fitness is evaluated by accuracy of prediction, number of folds in cross validation equals 5, and maximum time to run the code equals 10 minutes. The result shows that the accuracy is getting higher through generations and the average CV score on training data is 0.58. Then, I use the trained model to predict market trend in testing dataset. If today is predicted to be up-trending day, then buy 1 share of stock and sell tomorrow, otherwise do nothing. The cumulative returns of the strategy and the comparison with S&P 500 index shows below:



4. Optimization

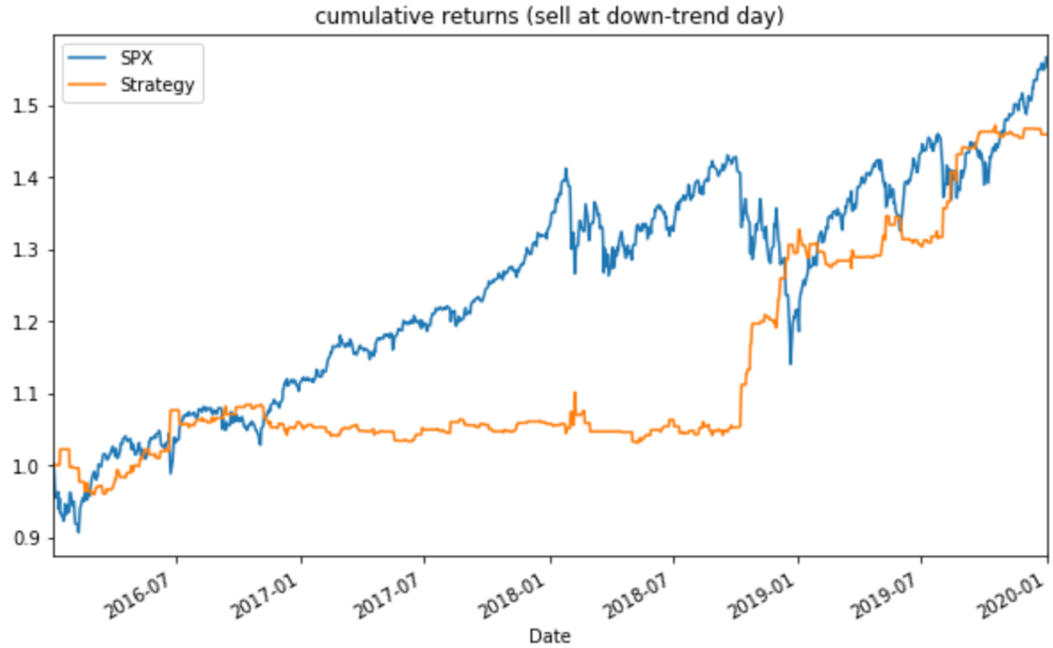
In order to select the best parameters and best combination of parameters, I use grid search and find out that the best strategy has 50 generations and population size 25. After that, I do the same calculation on this new strategy, and the cumulative returns turns out to be:



We can see that the cumulative return of this new strategy is much better than the former one.

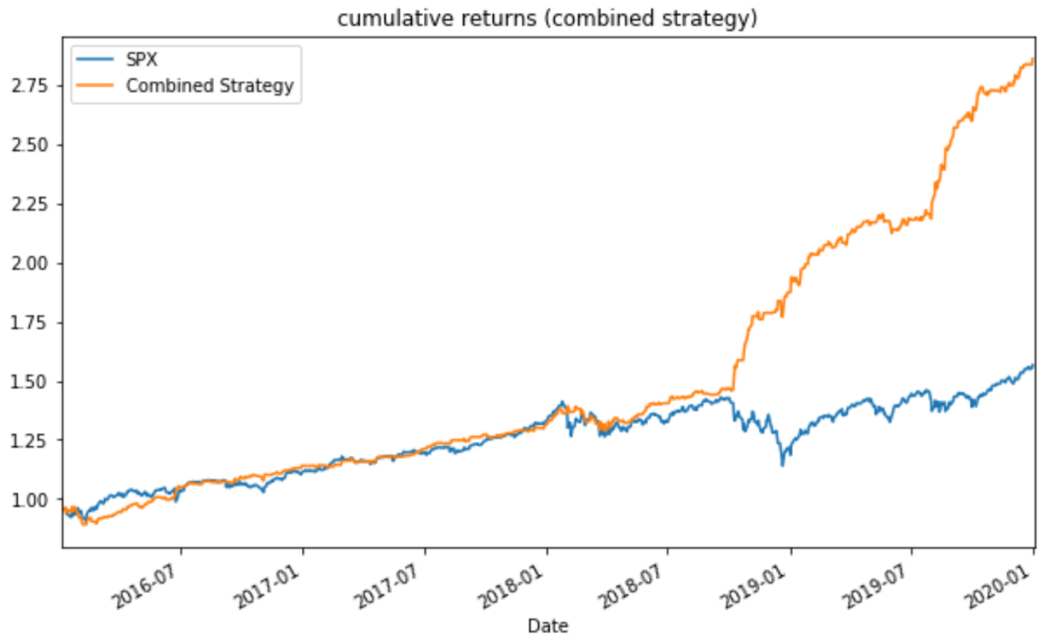
5. Find Down-trend Trading Days and Combine the Model

Now, I try to find the signal for down-trend trading days using the same approach and the best parameters in last model. The only thing different is the calculation for Hookday is different for up-trend and down-trend days. The average CV score on training set is 0.56 and the cumulative returns shows below:



The result is quite interesting. Although the total cumulative return of the strategy is lower than S&P500 index, but we can see that we the market goes down sharply between 2018-07 to early 2019, our strategy has large positive profit! To some aspect, the strategy successfully detects the down-trend signal of the market.

Finally, I combine the up-trend model and the down-trend model. If the up-trend model predicts today is up-trend day and the down-trend model predicts today is not a down-trend day, then buy the stock; if the up-trend model predicts today is up-trend day and the down-trend model also predicts today is a down-trend day, then do nothing; if the up-trend model predicts today is not a up-trend day and the down trend model predicts today is a down-trend day, then sell the stock; otherwise do nothing. The cumulative returns is as follows:



It's better than the former model on recent period, but not that good on precedent period, so the total cumulative return is similar.

6. Results and Conclusion

I use some technical features together with fuzzy membership functions to find out a trading strategy which can tell whether a trading day is a up-trend day or a down-trend day. The evolutionary algorithm used in the strategy is based on the assumption that the investment in the market is similar to a zero-sum game and we can always select the best trading strategy through generations. The empirical results on S&P 500 shows that evolutionary algorithm has the ability to detech the market trend and make good trading decisions.