MF 703

Programming for Mathematical Finance

Fall 2023

Course Project Report

Project Theme

Index Tracking using Machine Learning and Deep Learning Methods

----Exemplified on CSI 300

Team Members and Student ID

Ziyuan Zhang

Zhitong Ye

Yingzi Li

Yang Xia

Weichen Zhu

Abstract

This project conducts a comprehensive benchmark analysis over a five-year period, comparing various portfolio construction methods for half-year predictions utilizing one-year historical data. The findings reveal that Genetic Algorithm (GA) portfolios consistently outperform alternatives across seven distinct periods. Notably, GA portfolios exhibit superior performance in minimizing tracking error regardless of market trend. In contrast to the benchmark Sequential Least Squares Quadratic Programming (SLSQP) method, portfolios formed through deep learning and machine learning methods consistently demonstrate enhanced index tracking capabilities. This project try to contribute valuable insights into index tracking, particularly in the context of method selection. While GA portfolios emerge as robust performers, we advocate for cautious interpretation and underscores the importance of adaptability in addressing real-world trading challenges, hoping this project can serve as a reference for practices for index tracking.

Keywords: Index Tracking; Time-Weighted SVR; Deep Learning

1. Introduction

Developed from Markowitz's theory of portfolio investment, index investment is a kind of passive approach to build portfolios, which can outperform active ones in risk diversification, administrative expense control and etc[I]. Choosing index investment implies that the investor is comparably conservative and consider getting market average return with less risk as an acceptable choice[II].

In fact, considerable amounts of actively management funds do not achieve return greater than a typical benchmark, the market average level[i]. Over the last decade, transfer of a large amount of wealth from other funds to index funds has been witnessed, as passive management funds are supposed to have comparable advantage in lower trading and operating cost. To be specific, the capital scale under index funds' management over all funds management doubled from 2010 to 2020[III].

CSI 300, the index we tried to analyze, is regarded as a reflection of the whole Chinese stock market. Selected from listed A-share stocks mainly according to market value and liquidity, CSI 300 contains over half of the total market capitalization, excluding those ST, *ST or suspended onesIV, which can provide an overall reflection of the market trend. The amount of capital managed by passive funds tracking the CSI 300 reached 333 billion yuan by the end of May 2020V. In the last 5 years, the Chinese stock market has experienced much upheavals, it is worthy to review its performance for future investment.

Since indices like CSI 300 cannot be directly traded and a full replication is infeasible in practice, index tracking, creating a portfolio that invests in only a subset of the component stocks, to replicate a targeted return and risk is a common substitute[ii].

In aim of tracking the selected index, we try to minimize the tracking error between the portfolio and the index using different classical methods including machine learning and deep learning techniques, which are used frequently in index analysis in recent years. To be specific, we created portfolios using a genetic algorithm combined with SVM, LASSO, 2 kinds of neural network and a benchmark method Sequential Least Squares Programming (SLSQP) respectively.

As the CSI 300 index is reconstructed every June and December, our models consider a reform following every adjustment day to capture timely market information of risk and return. We compare the performance of these methods using benchmarks from several criterions, trying to provide some support for researches and practices on index tracking.

This paper is structured as follows. In Section 2, we describe the problem addressed in this paper. In Section 3, we present the methods used to construct the portfolio to replicate the CSI300 index. More specifically, we demonstrate how SLSOP, LASSO, time-weighted SVR, and Neural Networks can be applied to this index tracking problem in Section 3.1. In Section 3.2, we introduce the benchmarks used to evaluate the performance of the constructed portfolio. We then present full details of our experiments in Section 4. In Sections 4.1 and 4.2, we summarize the dataset and key experimental settings studied, and in Section 4.3, we analyze the index tracking performance of the proposed methods and discuss the results. Finally, in Section 5, we provide concluding remarks for this project, highlighting both the contributions and limitations of this study.

2. Problem description

The index tracking problem aims to minimize the tracking error[1] between a portfolio and the target index. This can be mathematically formulated as follows:

First introduce the data we used in tracking index. Denote $R_{i,t}$ as the daily return of i th stock, i = 1...N.

$$R_{i,t} = \frac{P_{i,t+1} - P_{i,t}}{P_{i,t}} \tag{1}$$

where: $P_{i,t}$ is the t^{th} day price of i^{th} stock, i = 1...N, t = 1...M.

When we minimize the tracking error, we trying to

$$\min \quad \text{Standard Deviation of (return of portfolio - return of underlying index)}^2 \tag{2}$$

And when doing quadratic optimization, we use the following constraints.

The first constraint is the capital budget constraint. ω_i represents the weight of the i^{th} stock in the investment portfolio.

$$\sum_{i=1}^{N} \omega_i = 1 \tag{3}$$

The second constraint is the upper and lower bounds on the investment ratio. It helps limit the ratio of weight of each stock in the investment portfolio to be greater than or equal to η_i and less than or equal to δ_i.

$$z_i \eta_i \leq \omega_i \leq z_i \delta_i, \; i = 1,2,\ldots,N \tag{4}$$

The third constraint is the limit on the total number of stocks entering the tracking portfolio, where the total number k is generally much smaller than the number N of all possible stocks in the market that can be used to track the index. The main purpose is to reduce the size, small management costs, and difficulty of the tracking portfolio.

$$\sum_{i=1}^{N} z_i = k \tag{5}$$

The fourth constraint is the constraint on variable z_i. When $z_i = 1$, it indicates that the i th stock enters the tracking portfolio, otherwise it is excluded. The value of this variable reflects the process of stock selection.

$$z_i = 0 \text{ or } 1, \; i = 1,2,\ldots,N \tag{6}$$

In this paper, we aim to solve two key issues: stock selection and portfolio weighting[3].

[1] Tracking error refers to the volatility of the deviation between the return on net worth of the ETF and the return on the underlying index

[2] This optimization problem can be formulated differently across various methods.

[3] Without further explanation, the notation retains the same meaning as shown in Section 2.

3. Method

3.1 Portfolio building

3.1.1 The baseline model : Sequential Least Squares Programming

To better evalute the peofomance of ourmodels, we try to find a baseline model for comparision. Here The Sequential Least Squares Programming (SLSQP) model is used to finish this purpose.

SLSQP formulizes index tracking as a constrained optimization problem. The objective is to minimize tracking error between the portfolio returns and index returns by optimizing over portfolio weights. Tracking error is measured by the sum of squared differences between the two return series.

The constraints are: Weights sum to one and individual weight bounds between 0 and 1. Given the market index I consisting of N stocks, the objective is to construct a portfolio P that closely replicates index performance over time.

Let R_t^I be the return of index I at time t, R_t^P be the return of portfolio P at time t.

$$\min_{\omega} \sum (R_t^I - R_t^P) \tag{7}$$

Subject to:

$$\sum_{i=1}^{N} \omega_i = 1 \tag{8}$$

$$0 \leqslant \omega_i \leqslant 1, \ \forall i \tag{9}$$

Where $R_t^I = \sum_{i=1}^{N} \omega_i R_{i,t}$

An initial guess of equal weights is set. Then SLSQP, a gradient-based solver, iteratively improves the weights by computing gradients and Hessians to minimize tracking error within constraints[VI].

Finally, the stocks with the top 30 highest optimized weights are selected to construct the tracking portfolio.

3.1.2 LASSO

Lasso (The Least Absolute Shrinkage and Selection Operator) is an effective method for variable selection in many application scenarios. Its core idea is to construct a primary penalty function, which compresses the coefficients of explanatory variables that have little impact on the dependent variable to zero, thereby achieving the purpose of variable selection.

$$\hat{\beta}^{\text{lasso}} = \arg\min_{\beta} \left(\frac{1}{2n} \|Y - X\beta\|_2^2 + \lambda \|\beta\|_1\right) \tag{10}$$

where:

Y represents the dependent variable.

X represents the matrix of independent variables.

β is the vector of coefficients.

n is the number of observations.

λ is the regularization parameter controlling the strength of the penalty.

Our methodology entails selecting the top thirty stocks by their respective weights as derived from the LASSO regression outcomes. After this selection, we normalize these weights to establish our definitive set of portfolio allocations. This step ensures that the sum of the weights is equal to one, adhering to the fundamental principle of portfolio construction[VII].

3.1.3 A genetic algorithm with a time-weighted SVR model

This method can be separated into two steps.

Firstly, we use a hybrid approach, which combines a genetic algorithm with quadratic optimization to select the stocks. Simply put, this genetic algorithm judges the quality of each portfolio by examining their quadratic optimization effects. This hybrid method can help us find the almost optimal tracking portfolio, and the computational cost is also relatively low.

Secondly, we use a time-weighted SVR model to re-optimize the weights of the selected stocks. SVR is a linear regression method based on SVM. It sets a small epsilon. If the deviation between the return of the tracking portfolio and the return of the target index does not exceed epsilon, we do not believe that the tracking portfolio deviates from the target index, meaning that the error is negligible. This can help avoid the overfitting problem. Time-weighted means giving recent data greater weight. This is because financial data sequences are unstable and have high noise, and data closer to the present has a stronger impact on the future, providing more information.

And finally, we get the following objective function.

$$\min \frac{1}{2}\|\omega\|^2 + C\sum_{t=1}^{T} \lambda_t L_\varepsilon(R_{mt}, R_{pt}) \tag{11}$$

where:

$\lambda_t = \frac{2}{1+e^{\alpha - 2\alpha t/T}},\ t = 1, 2, \ldots, T$ is the time-weighted parameter.

$L_\varepsilon(R_{mt}, R_{pt}) = \begin{cases} |R_{pt} - R_{mt}| - \varepsilon, & \text{if } |R_{pt} - R_{mt}| > \varepsilon \\ 0 \end{cases}$ .is the loss function.

According to the reasearch of CP Hu, HG Xue and FM Xu[VIII] and it's empirical analysis', we set C=20, α=1, ε=0.001.

3.1.4 NNF: Neural Network with Fixed noise

In recent years, artificial intelligence has become very popular, with the release of chatbots like ChatGPT, Claude, Llama, and Bard. Inspired by the Neural Network with Fixed Noise[IX], we implemented a simple neural network for stock selection and weighting to track market indices.

Given the relatively small amount of financial data compared to what is used to train language models, a 6-layer fully-connected network is sufficient. The key hyperparameters are:

- input_dim (300): The number of input features representing each stock's cumulative returns
- hidden_size1-5 (300): The number of nodes in each of the 5 hidden layers
- num_classes (300): The output nodes for predicted weights of each stock
- num_selection (30): The number of top stocks to select
- dropout_p (0.2): Dropout regularization probability

The network uses ReLU activations and dropout for the 5 hidden layers, with a softmax output layer to normalize the predicted stock weightings.

For the input data, we use cumulative returns. Denote $R^{i}_{cum}(t)$ as the cumulative returns of the i^{th} stock on day t.

$$R^{i}_{cum}(t) = \prod_{j=1}^{t}(1 + R_{i,t}) \tag{12}$$

Where $R_{i,t}$ is the Daily returns on day t of i^{th} stock.

In the NNF model, we take the last row of data as input features, representing cumulative returns of stocks up to the most recent period.

So the input data X comprises:

$$X = [R^{1}_{cum}(t_{end}),\ R^{2}_{cum}(t_{end}), ...,R^{N}_{cum}(t_{end})] \tag{13}$$

Where:

t_{end}: The last time period

N: the stock number

Then with this neural network, we use the output of the softmax layer to determine the portfolio weight of each stock.

$$softmax(\hat{y}_i) = \frac{e^{\hat{y}_i}}{\sum_{j=1}^{N} e^{\hat{y}_i}} \tag{14}$$

Where N is the size of the softmax layer, $\hat{y}_i$ is the i th output value of the previous layer of the softmax layer.

During training, the forward pass makes predictions, computes the MSE loss against the target cumulative returns, and updates the weights using backpropagation and Adam optimization.

Then we selecting the top 30 by weight for investment. Since the weights of a portfolio should sum to 1, we normalize these 30 stocks' weights to ensure their sum equals one, this is the nnf_partial model.

With further thinking, direct adjustment of weights might be too crude and potentially lose key information for tracking the index. Thus, we use neural network again, but this time , the stocks used to replicate is the selected 30 stocks is the former neural network. Then we view this updated weights as portfolio weights. This is the nnf model.

Considering the tracking error, we found that the nnf model, which replicates after stock selection, indeed yields better results with a smaller tracking error. The details of benchmarks can be seen in Table 2.

3.2 Evaluation benchmarks

8 Benchmarks from 4 criterions are selected for back-test period portfolios performance comparison. To present an overall performance in optimization, tracking error and market beta compared with CSI 300 index are calculated. Annualized volatility and max drawdown are used for risk management concern as index investment is a comparably conservative strategy. For risk-adjusted return, one can refer to sharp ratio and Sortino ratio. In aim of a pure return, annualized return and accumulated return are compared for persistent performance of models.

(Details in Appendix Ⅰ)

4.Experiment

4.1. Data description

CSI 300 compiled by China Securities Index Co., Ltd (CSI) since April 2005, is a basic index of two Chinese capital market, Shanghai Stock market and Shenzhen Stock market. There is a semi-annual adjustment of its constituents every June and December. As the market liquidity and the market size (the average daily turnover and average daily market capitalization) of these stocks over the past year is the main ranking basis, price data of a whole year are available for all ever-listed (selected as CSI 300 component) stocks.

We tried to look back the performance of CSI 300 over the last 5 years, thus CSI 300 data from 17 December 2018 to 12 June 2023 are under our concern. As a whole, CSI 300 increased rapidly around the first quarter of 2019, the second quarter of 2020 and the end of 2020. After peeked at 5807.72 in February 2021, it dropped gradually for about 2 years and become stable at around 3900 points in the first half of 2023. Compared to S&P 500, which rose continuously and rapidly from March 2020 to January 2022, CSI 300 stopped its upward trend after January 2021.

Graph 1 Daily Closing Price of CSI 300 and S&P 500

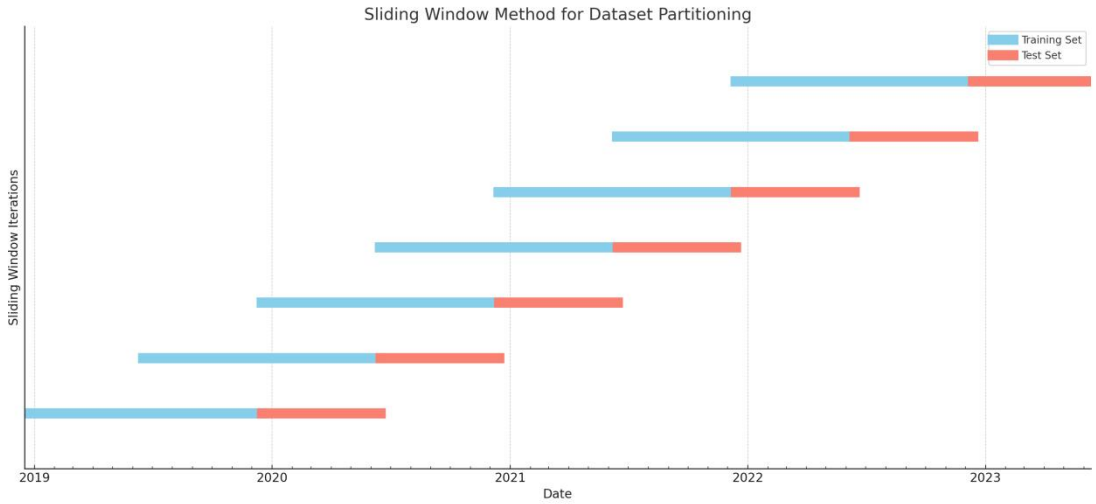
Our index component stock price data are obtained from China Securities Network4, which is the statutory information disclosure media of China Securities Regulatory Commission, and the entry and exit records of constituent stocks are from WIND database.

4.2. Experimental setting

As the CSI 300 component stocks are chose based on liquidity and the market size, they are adjusted every half year, in June and December. Therefore, we take the day after the adjustment of constituent stocks as the base date. A training set contains data of former year and a related test set contains data for a latter half year .

Through this method, our training and testing cycles are carried out in a looping manner until the entire period from December 17, 2018 to June 12, 2023 is fully covered.

[4] https://www.csindex.com.cn/#/indices/family/detail?indexCode=000300



Graph 2 Sliding Window Method for Dataset Partitioning

Table 1 Training periods and back-test periods

Periods	Training Period	Back-test Period
1	20181216 - 20191215	20191216 - 20200613
2	20190616 - 20200614	20200615 - 20201212
3	20191215 - 20201213	20201214 - 20210613
4	20200614 - 20210614	20210615 - 20211211
5	20201213 - 20211212	20211213- 20220611
6	20210614 - 20220612	20220613 - 20221210
7	20211212 - 20221211	20221212 - 20230610

To determine the composition of the portfolio we build, we use 4 methods mentioned above and regard minimizing the tracking error between the portfolio and the index as our objective function. Firstly, we select about 30 stocks from 300 component stocks of CSI 300 according to their performance in a training period and then we optimize the portfolio with a weight adjustment from a equally-weighted base.

In ignorance of related fees, we set a totally new portfolio every base date and do not make further adjustment for it over the latter back-test period. Then according to CSI 300 index returns and selected stocks' returns in the back-test period, we compare our portfolio's performance with that of CSI 300 using 8 benchmarks considering different criteria.

4.3. Performance evaluation

The portfolios' performance evaluation is carried mainly based on 8 benchmarks from 4 criterions.

Overall, the GA[5] portfolios performance best in pursuing a minimal tracking error. It has a stable small value of tracking error over all 7 back-test periods. Meanwhile, Lasso portfolios, NNF portfolios and NNF_Partial portfolios also performance better than the benchmark SLSQP Portfolio.

In the aspect of risk control, all these portfolios except SLSQP portfolios hold a moderate annualized volatility of around 0.20. In contrast, SLSQP portfolios only expose to an annualized

[5] Refer to the third method: A hybrid approach with a time-weighted SVR model

volatility half that amount. Consistent with volatility, the max Drawdown of returns of these portfolios present a similar characteristic.

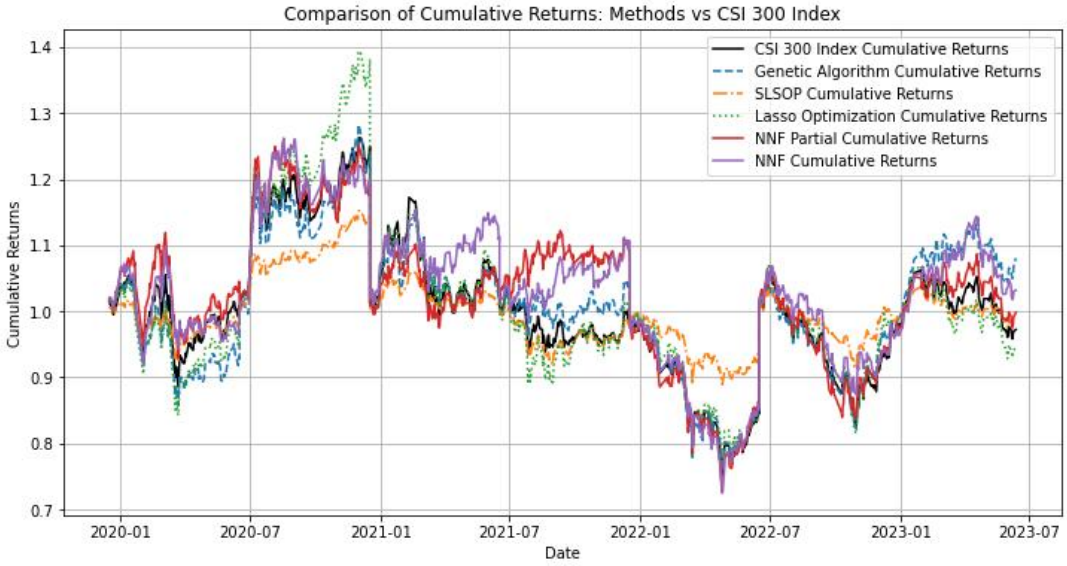
In the aspect of risk-adjusted returns, GA portfolios do not perform best in sharp ratio with an average of 0.1881, much less than all other portfolios except SLSQP. While Lasso portfolios, NNF_Partial portfolios and SLSQP portfolios' performance are much less stable, their leading positions mainly derive from a sharp ratio of more than 2 in the second back-test period. Only GA portfolios get a positive value of sharp ratio in each back-test period. Besides, it is suggested to choose methods other than a basic SLSQP to form a portfolio when downside risk is a great concern, as they can perform better in getting a higher Sortino ratio.

Though a higher return is not our standard to form portfolios, we compare their performance for reference. In consistent with their difference in market beta, all portfolios other than SLSQP are much sensitive to a bull or bear market. Additionally, our portfolios can support a higher return in cases that investors hold a right prediction of market trend. Even though short selling is unavailable in our portfolios for the convenience of model building, it can actually carry out through trading index futures or options.

Table 2 Performance Metrics of Benchmark Portfolios Across Multiple Periods

Benchmark	Portfolios	period 1	period 2	period 3	period 4	period 5	period 6	period 7	Mean
Tracking Error	GA	0.2782%	0.4107%	0.4801%	0.4422%	0.3347%	0.3340%	0.2832%	0.3662%
	Lasso	0.3285%	0.3664%	0.4229%	0.5456%	0.4188%	0.4145%	0.2882%	0.3978%
	NNF	0.3654%	0.4414%	0.5494%	0.5898%	0.6054%	0.5402%	0.3504%	0.4917%
	NNF_Partial	0.6314%	0.3825%	0.5237%	0.5367%	0.5492%	0.4516%	0.4505%	0.5037%
	SLSQP	0.8418%	0.7861%	0.8236%	0.5404%	0.7445%	0.5712%	0.4597%	0.6810%
Market Beta	GA	0.9696	0.9643	0.8048	0.8964	0.9423	0.9925	0.9101	0.9257
	Lasso	0.9929	1.0219	0.9874	1.1781	1.0840	1.1753	1.0420	1.0688
	NNF	0.9725	0.9677	0.7415	0.7875	0.9254	0.9681	0.9667	0.9042
	NNF_Partial	1.0872	0.9569	0.9776	0.8810	1.0140	0.9953	0.9314	0.9776
	SLSQP	0.4560	0.4348	0.4024	0.5770	0.5067	0.5348	0.4873	0.4856
Annualized Volatility	GA	0.2334	0.2138	0.1797	0.1586	0.2175	0.1865	0.1272	0.1881
	Lasso	0.2405	0.2232	0.2169	0.2056	0.2516	0.2197	0.1443	0.2146
	NNF	0.2371	0.2158	0.1688	0.1520	0.2280	0.1942	0.1386	0.1906
	NNF_Partial	0.2751	0.2108	0.2204	0.1628	0.2436	0.1931	0.1414	0.2068
	SLSQP	0.1114	0.0959	0.0900	0.1054	0.1187	0.1008	0.0688	0.0987
Max Drawdown	GA	16.4649%	6.8318%	10.1875%	8.4750%	20.9857%	20.6194%	10.8200%	13.4834%
	Lasso	19.3021%	6.8219%	13.0373%	13.2664%	22.6297%	23.9778%	13.1674%	16.0289%
	NNF	13.4310%	8.9370%	11.0261%	6.5750%	26.9488%	17.9533%	12.9146%	13.9694%
	NNF_Partial	17.7199%	6.6245%	15.9090%	7.4563%	26.8703%	16.3259%	12.8015%	14.8153%
	SLSQP	8.9840%	2.9311%	5.8078%	8.8334%	10.9621%	11.6727%	5.9317%	7.8747%
Sharpe Ratio	GA	0.2334	0.2138	0.1797	0.1586	0.2175	0.1865	0.1272	0.1881
	Lasso	-0.2055	3.0569	0.3731	-0.0059	-1.0595	-0.0208	-0.7717	0.1952
	NNF	-0.0020	1.4549	0.2678	1.5094	-1.0271	0.0645	-0.7302	0.2196
	NNF_Partial	0.7761	2.4725	0.2462	1.1712	-0.8938	-0.1372	-0.8475	0.3982
	SLSQP	-0.2475	2.9866	0.1798	-0.4177	-1.0303	0.1025	-0.4553	0.1597
Sortino	GA	2.3174	58.8669	8.3068	0.5309	-20.6966	-4.4330	-11.5634	4.7613

	Lasso	-4.1460	83.4300	8.4653	-0.1293	-21.8616	-0.5245	-17.6386	6.7993
	NNF	-0.0409	36.6135	5.7799	35.0973	-20.4007	1.5653	-15.8604	6.1077
	NNF_Partial	16.3515	64.1950	5.6050	26.7599	-17.9085	-3.2743	-18.1743	10.5078
	SLSQP	-5.0315	82.9118	4.0757	-9.0609	-20.9371	2.6150	-10.7703	6.2575
Cumulative Return	GA	-0.0288%	23.4785%	2.3055%	-0.4241%	-10.8924%	-2.4601%	-3.3447%	/
	Lasso	-3.6986%	38.1661%	2.7797%	-1.0976%	-13.1404%	-1.3972%	-5.4480%	/
	NNF	-1.3702%	15.3779%	1.4890%	11.4119%	-11.5696%	-0.3031%	-4.9705%	/
	NNF_Partial	8.7867%	27.8216%	1.4473%	9.1996%	-11.0326%	-2.2060%	-5.7984%	/
	SLSQP	-1.6074%	14.8682%	0.5813%	-2.4292%	-5.9253%	0.2635%	-1.5372%	/
	GA	-0.0397%	33.6983%	3.2063%	-0.5835%	-14.7585%	-3.3901%	-4.7807%	1.91%
	Lasso	-5.0574%	56.0772%	3.8692%	-1.5083%	-17.7215%	-1.9293%	-7.7501%	3.71%
Annualized Return	NNF	-1.8819%	21.7715%	2.0676%	16.0451%	-15.6541%	-0.4194%	-7.0786%	2.12%
	NNF_Partial	8.7867%	27.8216%	1.4473%	9.1996%	-11.0326%	-2.2060%	-5.7984%	4.03%
	SLSQP	-2.2067%	21.0314%	0.8058%	-3.3297%	-8.1096%	0.3651%	-2.2061%	0.91%



Graph 3 Comparision of Cumulative Returns-Methods vs CSI 300 Index

We use the cumulative return to demonstrate the fitting results of our models. Graph 3 shows a comparison of "Cumulative Returns - Methods vs. CSI 300 Index" over time. The plot includes multiple lines, each representing a different investment strategy. The solid black line represents the CSI 300 Index Cumulative Returns, which serves as our benchmark reference.

It can be seen that different strategies have different performances over various time periods. However, overall, GA portfolios perform the best, with relatively small deviations in all time frames. This is logical, as this model accounts for the time effect, meaning it gives more weight to data points that are closer to the node.

Additionally, the Lasso method, indicated by the green line, also appears to capture the trend of the index changes quite well. Its advantage is that it does not fall much during downturns, and when it exceeds the index return rate, it is comparatively higher. This might be because the Lasso method selects stocks that have the most significant impact on the market. In the market, the stock price movements of large companies can largely reflect the overall market trends. Thus, when the market declines, people tend to buy stocks of large companies for safety, causing them not to fall

much. Conversely, when the market rises, people lean towards investing in large companies due to their clear growth momentum.

5. Conclusion

Through benchmark comparison, we find GA portfolios perform best in a half year prediction with a whole year past data over 7 periods in the last 5 years. Among these portfolios formed using different methods of stock selection and weight determination, GA portfolios have a persistent fine performance in minimizing tracking error, whenever the market trend is upward, downward or fluctuating. Compared with a benchmark method of SLSQP, all these portfolios formed using deep learning and machine learning methods are better in index tracking.

There is much limitation in our project deserve further research. As fixed training periods of 1 year, a following back-test period of half year and portfolio decision each period is set in our research based on regular index component change of CSI, flexibility is a great concern for real trade. Though we set base day as the day of component adjustment and get relevant fine optimization results, there is some doubt about the announcement of stocks confirmation as component of CSI 300. As their stock performance may have been positively influence long before actual adjustment, the chance of their selection into our portfolios may be elevated with miss leads. In prudence, related researches need to be carried out using different indices for different training and back-test period combinations.

All in all, we suppose our research can provide some reference for index tracking when methods selection is a concern.

Appendix

I Evaluation benchmarks

Benchmark 1: Tracking Error

Tracking Error is the standard deviation of the difference in returns between a portfolio and its benchmark index. It measures how much the portfolio's performance deviates from the index. Ryan(1998) believed that tracking error was an effective risk measurement method as it could measure the relative risk of portfolio in achieving investors' real investment objectives.

$$\text{Tracking Error} = \sqrt{\frac{\sum_{t=1}^{n}(R_{pt} - R_{mt})^2}{n-1}}$$

where R_{pt} is the portfolio return at time t and R_{mt} is the benchmark index return at time t.

Benchmark 2: Market Beta

Market Beta is a measure of the portfolio's sensitivity to overall market movements, specifically to the returns of a chosen market index or benchmark. A beta greater than 1 indicates higher volatility than the market, and less than 1 indicates lower.

$$\text{Market Beta} = \frac{\text{cov}(R_{pt}, R_{mt})}{\text{var}(R_{mt})}$$

where R_{pt} is the portfolio return at time t and R_{mt} is the benchmark index return at time t.

Benchmark 3: Annualized Volatility

Annualized Volatility measures the standard deviation of investment returns and scales it to a yearly measure. It is a valuable metric for ETF investors as it aids in assessing risk, making informed investment decisions, and optimizing portfolio construction for a better risk-return trade-off.

$$\text{Annualized Volatility} = \sigma_p * \sqrt{252}$$

where σ_p is the standard deviation of daily returns and 252 is the typical number of trading days in a year.

Benchmark 4: Max Drawdown

Maximum Drawdown measures the maximum risk of loss that a portfolio or strategy can face over a period of time. Specifically, it is the maximum cumulative decline in a portfolio from peak (high point) to trough (low point). The significance of Maximum Drawdown is to help investors more fully understand the risks and potential losses of a portfolio or strategy, so as to better manage and optimize their investment decisions.

$$\text{Maximum Drawdown} = -\frac{\text{lowest cummulative return} - \text{highest cummulative return}}{\text{highest cummulative return}}$$

Benchmark 5: Sharpe Ratio

Sharpe Ratio is a measure of risk-adjusted return, which is the excess return earned per unit of volatility or total risk. Using the Sharpe Ratio as a benchmark for index replication provides a

clear and quantifiable measure of how effectively a portfolio or investment strategy is generating returns relative to the risk it carries.

$$\text{Sharpe Ratio} = \frac{R_{ap} - R_{af}}{SD_p}$$

where R_{ap} is the annualized average return, R_{af} is the annualized risk-free rate, SD_p is the annualized standard deviation (volatility) of excess return.

Benchmark 6: Sortino Ratio

Sotino ratio is a measure of the relative performance of a portfolio. There are similarities with the Sharpe ratio, but the Sotino ratio uses the lower partial standard deviation instead of the total standard deviation to distinguish between unfavorable and favorable fluctuations. Similar to the Sharpe ratio, the higher the ratio, the higher the excess rate of return for the fund taking the same unit of downside risk.

$$\text{Sortino Ratio} = \frac{R_{ap} - \text{MAR}}{\text{DR}}$$

where R_{ap} is the annualized average return, MAR is min acceptable return, DR is the downward standard deviation. MAR can be the risk-free rate, it can be 0, or it can be the level of return that other investors agree on.

Benchmark 7: Annualized Return

Annualized Return converts the return of a portfolio to an annual figure, allowing comparison between investments over different lengths of time. Annualized Return provides a clear, time-adjusted measure of how well a fund is tracking its benchmark index over an extended period, which is essential for investors making long-term investment decisions.

$$\text{Annualized Return} = (1 + \text{Cumulative Return})^{\frac{252}{D}}$$

where $(1 + \text{Cumulative Return})$ is the factor by which the initial investment has grown. $\frac{252}{D}$ is the factor that scales the return to an annual (252-day) basis.

Benchmark 8: Cumulative Return

Cumulative Return is the total rate of return on an investment over a set time period. Cumulative Return serves as a holistic measure for investors to evaluate how well an index fund or ETF has achieved its objective of tracking the performance of the benchmark index over time. It helps in determining whether the portfolio has delivered the expected investment growth in line with the index it aims to replicate.

$$\text{Cumulative Return} = \prod_{t=1}^{n} (1 + R_t)$$

where R_t is the returns for period t.

Reference

I Yang, T., & Huang, X. (2022). Two new mean − variance enhanced index tracking models based on uncertainty theory. The North American Journal of Economics and Finance, 59, 101622. doi:10.1016/j.najef.2021.101622

II Ruiz-Torrubiano, R., & Suárez, A. (2009). A hybrid optimization approach to index tracking. Annals of Operations Research, 166(1), 57-71. doi:10.1007/s10479-008-0404-4

III Adams, J., Hayunga, D., & Mansi, S. (2022). Index fund trading costs are inversely related to fund and family size. Journal of Banking & Finance, 140, 106527. doi:10.1016/j.jbankfin.2022.106527

IV Chu, G., Goodell, J. W., Li, X., & Zhang, Y. (2021). Long-term impacts of index reconstitutions: Evidence from the CSI 300 additions and deletions. Pacific-Basin Finance Journal, 69, 101651. doi:10.1016/j.pacfin.2021.101651

V Fang Liu, & Wen Wen. (2023). Stock index constituent adjustments and corporate social responsibility: Evidence from a quasi-natural experiment of CSI 300 index adjustments. China Journal of Accounting Studies, 11(2), 1-32. doi:10.1080/21697213.2023.2239662

VI Wu, L., Wang, Y., & Wu, L. (2022). Modeling index tracking portfolio based on stochastic dominance for stock selection. The Engineering Economist, 67(3), 172-194. doi:10.1080/0013791X.2022.2047851

VII Sant'Anna, L. R., Caldeira, J. F., & Filomena, T. P. (2020). Lasso-based index tracking and statistical arbitrage long-short strategies. The North American Journal of Economics and Finance, 51, 101055. doi:10.1016/j.najef.2019.101055

VIII Hu, C. P. , Xue, H. G. , & Xu, F. M. . An stock index replicating model based on time weighted svm and it's empirical analysis. Systems Engineering-Theory & Practice.

IX Kwak, Yuyeong, Junho Song, and Hongchul Lee. "Neural network with fixed noise for index-tracking portfolio optimization." Expert Systems with Applications 183 (2021): 115298.