High-dimensional portfolio optimisation

Autumn 2024

M.Stat 1st Year

Indian Statistical Institute, Kolkata

Multivariate Analysis Project

Debabrata Sarkar (MB2309)

(Group K)

Supervisor:

Prof. Minerva Mukhopadhyay

Interdisciplinary Statistical Research Unit

Indian Statistical Institute

Contents

1 Abstract

When a portfolio consists of a large number of assets, it is accompanied by many small and illiquid positions that need frequent rebalancing. Avoiding such unstable portfolios is essential when it comes to financial index tracking because they are expensive and difficult to manage. *Partial replication* solves this problem by building a sparse monitoring portfolio with a smaller number of assets. Traditionally, this is done by imposing a cardinality constraint, which directly restricts the amount of assets in the portfolio. Unfortunately, this approach necessitates pre-specifying the maximum number of assets that can be chosen, which is rarely possible. Furthermore, the cardinality-constrained optimization problem is NP-hard. Solving this problem will be computationally expensive, especially in high-dimensional settings.

In the paper *High-dimensional index tracking based on the adaptive elastic net*[1], the authors propose a regularisation approach based on an extension of Lasso and Elastic Net models, called the Adaptive Elastic Net (Aenet) and provide a coordinate descent algorithm for the same.

2 Objective function

2.1 The index tracking problem

One of the most commonly used tracking error measures is the mean squared error. Under the full investment and no-short selling constraints, the index tracking problem is defined as:

$$\min_{w} \frac{1}{n}\|y - Xw\|_2^2, \tag{1}$$

subject to:

$$\sum_{j=1}^{p} w_j = 1, \quad w_j \geq 0. \tag{2}$$

This full model performs index tracking by generating a full replication of the target portfolio. However, as stated before, this is expensive to execute and is NP-hard for high-dimensional data. Thus, we want to only partially replicate the target portfolio, which has fewer assets. This is essentially the same as introducing sparsity to the portfolio. A natural choice for introducing sparsity is the Lasso estimator.

2.2 Lasso

The **Lasso** estimator proposed by Tibshirani (1996) [2] is given by:

$$\hat{\beta}_{\text{Lasso}} = \arg\min_{\beta} \|y - X\beta\|_2^2 + \lambda\|\beta\|_1. \tag{3}$$

However, Lasso has some limitations.

- The Lasso estimator has a noticeably large bias.
- Lasso could be inconsistent for variable selection unless the predictor matrix satisfies a rather strong condition.
- The presence of collinearity, which is often encountered in high-dimensional data, makes the Lasso paths unstable.

2.3 Adaptive Lasso

To overcome the inconsistency issue, Zou (2006) [3] proposed the following **Alasso** estimator:

$$\hat{\beta}_{\text{Alasso}} = \arg\min_{\beta} \|y - X\beta\|_2^2 + \lambda \sum_{j=1}^{p} v_j |\beta_j|, \tag{4}$$

where $v_j = (|\hat{\beta}_j^{\text{init}}|)^{-\tau}$ is an initial consistent estimate of β.

2.4 Elastic net

To stabilize the solution paths, Zou and Hastie (2005) [4] proposed the Elastic Net (**Enet**) estimator by introducing an extra L_2 penalty as seen in Ridge regression, which reduces the effect of multi-collinearity.

$$\hat{\beta}_{\text{Enet}} = \arg\min_{\beta} \|y - X\beta\|_2^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \|\beta\|_2^2. \tag{5}$$

The Alasso and Enet estimators improve Lasso in two different directions:

- The Alasso overcomes the inconsistency problem.
- The Enet improves the stability of the solution paths.

2.5 Adaptive elastic net

Zou and Zhang (2009) [5] combined the ideas of the Alasso and Enet in order to obtain an even better method, the Adaptive Elastic Net (**Aenet**).

$$\hat{\beta}_{\text{Aenet}} = \left(1 + \frac{\lambda_2}{n}\right) \left[\arg\min_{\beta} \|y - X\beta\|_2^2 + \lambda_1 \sum_{j=1}^{p} v_j |\beta_j| + \lambda_2 \|\beta\|_2^2\right], \tag{6}$$

where $v_j = (|\hat{\beta}_j^{\text{init}}|)^{-\tau}$ has the same meaning as before.

3 Sparse index tracking with Aenet penalty

The sparse index tracking problem with the Aenet penalty can be formulated as:

$$\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} x_{ij}w_j\right)^2 + \lambda_1 \sum_{j=1}^{p} v_j|w_j| + \lambda_2\|\mathbf{w}\|_2^2 + \lambda_c \sum_{j=1}^{p}|w_j - \bar{w_j}|, \quad (7)$$

subject to:

$$\mathbf{w}^\top \mathbf{1_p} = 1, \quad w_j \geq 0, \quad (8)$$

where $\mathbf{e}$ is the ($p \times 1$) vector of 1's.

Constraints in (8) are the budget and no-short selling constraints, respectively. The L_1 weighted penalty controls the sparsity of the portfolio weights. The L_2 weighted penalty reduces the multicollinearity in the predictors. The term $\lambda_c \sum_{j=1}^{p} |w_j - \bar{w}|$ is the turnover penalty, where w_j is portfolio weight of asset j in the previous time period. We set the adaptive weight $v_j = (|\hat{\beta}_j^{\text{init}}|)^{-\tau}$ as the solution to index tracking problems (1)–(2).

To optimise this loss function, we shall make use coordinate-descent algorithm using sub-gradients.

4 Coordinate descent algorithm

The coordinate descent algorithm iteratively optimises one coordinate (variable) at a time while holding the others constant in order to minimise (or maximise) an objective function. When the objective function is separable, it works especially well for high-dimensional problems.

Let $\mathbf{w} = (w_1, w_2, \ldots, w_p)$ be a vector of p variables, and $f((w))$ be the objective function to be minimized. The coordinate descent algorithm updates one variable at a time, keeping the others fixed.

The update rule for the j-th coordinate at iteration $k+1$ is:

$$w_j^{(k+1)} = \arg\min_{w_j} f\left(w_1^{(k+1)}, \ldots, w_{j-1}^{(k+1)}, w_j, w_{j+1}^{(k)}, \ldots, w_p^{(k)}\right),$$

where k denotes the iteration index.

5 Problem formulation

The sparse index tracking objective function using the Adaptive Elastic Net (Aenet) penalty can be expressed as:

$$\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} x_{ij}w_j\right)^2 + \lambda_1 \sum_{j=1}^{p} v_j|w_j| + \lambda_2\|\mathbf{w}\|_2^2 + \lambda_c \sum_{j=1}^{p}|w_j - w_j^{\text{prev}}|, \quad (9)$$

subject to:

$$\mathbf{w}^\top \mathbf{e} = 1, \quad w_j \geq 0, \tag{10}$$

where:

- y_i is the return of the target index at time i.
- x_{ij} is the return of asset j at time i.
- $\mathbf{w} = [w_1, w_2, \ldots, w_p]^\top$ is the vector of portfolio weights.
- λ_1, λ_2, and λ_c are regularization parameters for sparsity, stability, and turnover penalties, respectively.
- $v_j = (|\hat{\beta}_j^{\text{init}}|)^{-\tau}$ is the adaptive weight for the j-th asset.
- w_j^{prev} is the previous portfolio weight for asset j.

5.1 Propositions with explanations

We state two propostions which will help us simplify the optimisation problem (refer to [1] for proof).

5.1.1 Effect of No-Short Selling Constraints

The solution to the optimization problem:

$$\min_{\mathbf{w}} \frac{1}{n} \|y - X\mathbf{w}\|_2^2, \quad \text{s.t.} \quad \mathbf{w}^\top \mathbf{e} = 1,\ w_j \geq 0,\ \forall j$$

is equivalent to the solution of:

$$\min_{\mathbf{w}} w^\top A\mathbf{w} - 2B^\top \mathbf{w}, \quad \text{s.t.} \quad \mathbf{w}^\top \mathbf{e} = 1,\ w_j \geq 0,\ \forall j$$

where:

- A is the covariance matrix of asset returns, $A = \frac{1}{n} X^\top X$, where X is the $n \times p$ matrix of asset returns.
- B is the correlation vector, $B = \frac{1}{n} X^\top y$, where y is the $n \times 1$ vector of index returns.
- $\tilde{A} = A - \frac{\gamma \mathbf{e}^\top + \mathbf{e}\gamma^\top}{2}$ is the modified covariance matrix under the no-short selling constraint.
- γ is the vector of Lagrange multipliers for the non-negativity constraints $w_j \geq 0$.

5.1.2 Effect of Adaptive Elastic-Net Penalty

The solution to the optimization problem with the adaptive elastic-net penalty:

$$\min_{\mathbf{w}} \frac{1}{n}\|y - X\mathbf{w}\|_2^2 + \lambda_1 \sum_{j=1}^{p} v_j|w_j| + \lambda_2\|\mathbf{w}\|_2^2 + \lambda_c \sum_{j=1}^{p} |w_j - w_j^0|,$$

subject to:

$$\mathbf{w}^\top \mathbf{e} = 1,\ w_j \geq 0,\ \forall j,$$

is equivalent to the solution of:

$$\min_{\mathbf{w}} \mathbf{w}^\top A\mathbf{w} - 2B^\top \mathbf{w}, \quad \text{s.t.} \quad \mathbf{w}^\top \mathbf{w} = 1,\ w_j \geq 0,\ \forall j,$$

where:

- $\tilde{A} = A - \frac{\gamma e^\top + e\gamma^\top}{2} + \lambda_2 I$ is the modified covariance matrix.
- $\tilde{B} = B - \frac{1}{2}\lambda_c \mathbf{g} - \frac{1}{2}\lambda_1 \mathbf{v}$ is the modified correlation vector.
- $\mathbf{g}$ is the subgradient vector of the turnover penalty term $\sum_{j=1}^{p} |w_j - w_j^0|$, where w^0 represents the weights from the previous time period.
- $\mathbf{w}^\top \mathbf{e} = 1$ ensures full investment, where the weights sum to 1.
- $w_j \geq 0$ enforces the no-short selling constraint.

5.2 Constraints as penalties

To enforce the constraints, we usually introduce penalty terms as follows:

$$\mathcal{P}(\mathbf{w}) = \alpha \left(\mathbf{w}^\top \mathbf{e} - 1\right)^2 + \beta \sum_{j=1}^{p} \max(0, -w_j), \tag{11}$$

where:

- $\alpha > 0$ is the penalty coefficient for the budget constraint.
- $\beta > 0$ is the penalty coefficient for the non-negativity constraint.

However, we can cleverly replace the penalties using the sub-gradients defined as:

$$g_1(\mathbf{w}) = \begin{cases} 0, & \text{if } w'e = 1, \\ \infty, & \text{otherwise,} \end{cases}$$

and

$$g_2(w_j) = \begin{cases} 0, & \text{if } w_j \geq 0, \\ \infty, & \text{otherwise.} \end{cases}$$

Let $f_0(w) = \frac{1}{2}\sum_{i=1}^n \left(y_i - \sum_{j=1}^p x_{ij}w_j\right)^2 + g_1(w)$ and $f_1(w_j) = \lambda_c|w_j - \bar{w}_j| + \lambda_1\hat{v}_j w_j + \lambda_2 w_j^2 + g_2(w_j)$. The objective function in can then be rewritten as:

$$\min_{\mathbf{w}} f(\mathbf{w}) = \min_{\mathbf{w}} \left\{ f_0(\mathbf{w}) + \sum_{j=1}^p f_j(w_j) \right\}. \tag{12}$$

Thus the objective is separably additive, and we can use the coordinate descent algorithm to optimise it.

5.3 Coordinate descent algorithm

$$\text{Let} \quad \tilde{y}_i^{(j)} = \sum_{k\neq j} x_{ik}\tilde{w}_k, \quad d_j = \sum_{i=1}^n x_{ij}\left(y_i - \tilde{y}_i^{(j)}\right), \quad c_j = \sum_{i=1}^n x_{ij}^2, \quad \text{and}$$

$$S_u^{(m)} = \{j : w_j^{(m)} > \bar{w}_j \geq 0\}, \quad S_m^{(m)} = \{j : w_j = \bar{w}_j > 0\}, \quad S_l^{(m)} = \{j : 0 < w_j^{(m)} < \bar{w}_j\}.$$

Algorithm 1 Coordinate Descent Update for the High-dimensional Index Tracking

Input: Regularization parameters λ_1, λ_2, λ_c, and τ.

Output: Portfolio weights $\mathbf{w}$.

Compute initial weights $\hat{\beta}^{\text{init}}$ and previous portfolio weights $\mathbf{w}^{\text{prev}}$.

Initialize $\mathbf{w}^{(0)} = \frac{1}{p}\mathbf{e}$, where $\mathbf{e}$ is a vector of ones.

Initialize $\gamma_0^{(0)} > \max_{1\leq j\leq p}\{v_j\lambda_1 + \lambda_c\}$, where $v_j = (|\hat{\beta}_j^{\text{init}}|)^{-\tau}$.

repeat

For each $j = 1, 2, \ldots, p$ and $m > 0$:

1. Compute:

$$w_j \leftarrow \begin{cases} \frac{d_j+\gamma_0-v_j\lambda_1-\lambda_c}{c_j+2\lambda_2}, & \text{if } w_j < \frac{d_j+\gamma_0-v_j\lambda_1-\lambda_c}{c_j+2\lambda_2}, \\ w_j, & \text{if } \frac{d_j+\gamma_0-v_j\lambda_1-\lambda_c}{c_j+2\lambda_2} \leq w_j \leq \frac{d_j+\gamma_0-v_j\lambda_1+\lambda_c}{c_j+2\lambda_2}, \\ \frac{d_j+\gamma_0-v_j\lambda_1+\lambda_c}{c_j+2\lambda_2}, & \text{if } w_j > \frac{d_j+\gamma_0-v_j\lambda_1+\lambda_c}{c_j+2\lambda_2}. \end{cases}$$

For $m > 0$, update the Lagrange multiplier:

$$\gamma_0 \leftarrow \left(\sum_{j\in S_u\cup S_l} \frac{1}{c_j+2\lambda_2}\right)^{-1} \left(1 - \sum_{j\in S_m} w_j - \sum_{j\in S_u\cup S_l} \frac{d_j - v_j\lambda_1}{c_j+2\lambda_2} - \sum_{j\in S_l} \frac{\lambda_c}{c_j+2\lambda_2} + \sum_{j\in S_u} \frac{\lambda_c}{c_j+2\lambda_2}\right).$$

until Convergence of $\mathbf{w}$ and γ_0.

Return the optimized portfolio weights $\mathbf{w}$.

6 Data Analysis

6.1 Data Description

The NIFTY 50 is an Indian stock market index that represents the float-weighted average of 50 of the largest Indian companies listed on the National Stock Exchange. The NIFTY 50 index ecosystem consists of index funds (both onshore and offshore mutual funds and ETFs), and futures and options at NSE and NSE International Exchange.

The **Yahoo Finance - Nifty 50 Historical Data** contains daily recorded values of the following variables for each index: Date, Open (opening price), High (maximum price attained), Low (minimum price attained), Close (closing price), Adj price (closing price adjusted for splits and dividends), and Volume (number of shares traded).

6.2 Cleaning the data

For our purpose, we shall use the data recorded between the time period 1st January, 2010 to 1st July, 2024. Any missing prices are imputed by the linear interpolation approach. We compute the daily log-return:

$$x_{t,j} = \log\left(\frac{P_{t,j}}{P_{t-1,j}}\right); \quad t = 1, 2, ..., T$$

where where T is the total number of periods in a data set, and $P_{t,j}$ is the daily price of asset j on day t.

The stocks 'SBILIFE.NS', 'HDFCLIFE.NS', 'COALINDIA.NS' were removed because of less stock trading days and 'BEL.NS' was removed because $\hat{\beta}_j^{\text{init}} = 0$. Along with the above, rows corresponding to non-trading days were also removed.

6.3 Data Analysis

The summary statistics of the data is given below:

Statistic	**Value**
Mean	0.00042883858885077779
Median	0.0007029866326317706
Standard Deviation (std)	0.010695179023719185
Minimum (min)	-0.13903756461051203
Maximum (max)	0.08400295216282225

Table 1: Nifty 50 Stats

6.4 Training and testing

To identify index tracking investing strategies, a moving time window approach was used in each experiment. Specifically, to identify the ideal tracking portfolio, a training window of size $T_{train}(< T)$ is initially chosen. The performance for the next T_{test} out-of-sample trading days is then assessed while it remains unchanged. We must redesign the new tracking portfolio at the conclusion of this testing phase. To do this, we forward the training window by T_{test} days, using the final T_{train} days for portfolio design and the following T_{test} days for portfolio evaluation.

We first calculate $\hat{\beta}_j^{\text{init}}$ and set v_j using $\tau = 1$ for simplicity. The number of features is set to $p = 46$. X is the matrix of log-returns. The target vector y represents the log-returns of the Nifty 50 index. The regularization parameters are initialized as $\lambda_1 = 0.01$, $\lambda_2 = 0.1$, $\lambda_c = 10^{-4}$ and $\gamma_0 = 115$. The vector of weights $\mathbf{w}$ is taken as the vector of ones. The convergence tolerance is set to 10^{-5}, and the maximum number of iterations for the algorithm is set to 10000. After convergence, $\gamma_0^{(10000)} = 0.3134398007826907$

6.5 Performance metrics

The out-of-sample returns at time t, where t = $T_{train}+1, ..., T_{train}+T_{test}$, based on the portfolio weights determined in the first training window $\mathbf{w_1}$ are given by:

$$r_t = \sum_{j=1}^{p} w_{1,j} \prod_{i=T_{train}+1}^{t} (1 + x_{i,j})$$

The performance of a tracking portfolio is assessed using the metrics defined below.

- **Tracking error (TE):** The tracking error measures how closely the tracking portfolio replicates the index.

$$\text{TE} = \sqrt{\frac{1}{T_{test}} \sum_{i=T_{train}+1}^{T_{train}+T_{test}} (r_t - y_t)^2}$$

- **Average active return (AR):** The active returns are the investment returns on the tracking portfolio that exceed the returns on the underlying index. Positive active returns show that the tracking portfolio outperforms the index, while negative active returns show that the tracking portfolio underperforms the index.

$$\text{AR} = \frac{1}{T_{test}} \sum_{i=T_{train}+1}^{T_{train}+T_{test}} (r_t - y_t)$$

- **Portfolio turnover (TO):** The turnover measures the stability of the tracking portfolio. Lower turnover means lower transaction cost.
  Let N = $(T - T_{train})/T_{test}$ and $w_{i,j}$ be the desired weight of asset j at the i-th window (after rebalancing).

$$\text{TO} = \frac{1}{N-1} \sum_{i=1}^{N-1} \sum_{j=1}^{p} |w_{i+1,j} - w_{i,j}|$$

The general way to select the regularization parameters in index tracking is minimizing the tracking error by using the K-fold cross validation.

6.6 Results

Set $T_{train} = 250$ and $T_{test} = 21$. The summary of the metrics is given below. After cross-validation, the final weighted vector after convergence is given below.

Metric	Mean	Standard Deviation
Active Return (AR_os)	0.04999	0.51581
Tracking Error (TE_os)	0.36333	0.37049
Turnover	0.0	0.0

Table 2: Summary of Active Return (AR), Tracking Error (TE), and Turnover

```
[29.08897377 32.65511121 58.52551128 86.50954108  9.14001724
    56.1740587
  0.          0.          7.74848073 24.91202191 46.9220122
    44.78637566
 35.0847694   0.          0.         23.0629624   0.          0.
  0.         24.18428584  0.          0.         14.08875943
    18.24564754
  0.          0.          0.          0.          0.
    12.80292457
  0.          0.         19.1966936   0.          0.          0.
  0.          8.32272742  0.          0.          0.          0.
  0.          0.          0.          0.        ]
```

The final weight vector has only 18 non-zero entries. Thus, sparse index-tracking replication has been achieved.

6.7 Visual metrics

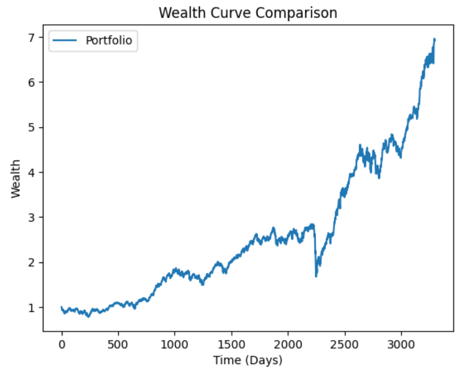

Figure 1: Tracking portfolio performance over time

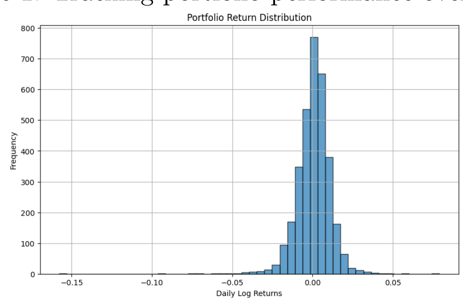

Figure 2: Tracking portfolio log retruns

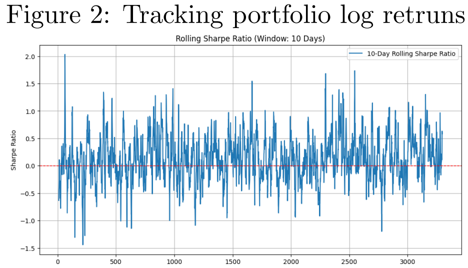

Figure 3: Tracking portfolio Sharpe Ratio

The following curves compare the tracking portfolio against Nifty50. As

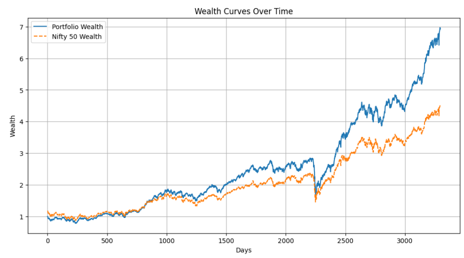

Figure 4: Tracking portfolio vs Nifty50 returns

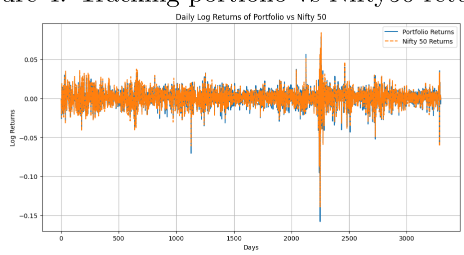

Figure 5: Tracking portfolio vs Nifty50 Sharpe Ratio

is apparent from the comparison curves, which has outperformed the original portfolio for the given out-of-sample data. Thus, the sparse index tracking portfolio is excellent.

6.8 Other financial metrics

The values of some other important financial metrics used to judge the performance of a portfolio are given below.

Metric	**Value**
Max Drawdown	-41.19%
Max Drawdown Duration	421 days
Alpha	-0.0012
Correlation with Nifty 50	0.9558
Sharpe Ratio	-2.7510

Table 3: Performance Metrics Summary

References

[1] Lianjie Shu, Fangquan Shi, and Guoliang Tian. "High-dimensional index tracking based on the adaptive elastic net". In: *Quantitative Finance* 20.9 (2020), pp. 1513–1530.

[2] Robert Tibshirani. "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 (1996), pp. 267–288. ISSN: 00359246. URL: `http://www.jstor.org/stable/2346178` (visited on 11/17/2024).

[3] Hui Zou. "The adaptive LASSO ad its oracle properties". In: *Journal of the American Statistical Association* 101 (Feb. 2006), pp. 1418–1429. DOI: `10.1198/016214506000000735`.

[4] Hui Zou and Trevor Hastie. "Zou H, Hastie T. Regularization and variable selection via the elastic net. J R Statist Soc B. 2005;67(2):301-20". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67 (Apr. 2005), pp. 301–320. DOI: `10.1111/j.1467-9868.2005.00503.x`.

[5] Hui Zou and Hao Helen Zhang. "On the adaptive elastic-net with a diverging number of parameters". In: *The Annals of Statistics* 37.4 (Aug. 2009). ISSN: 0090-5364. DOI: `10.1214/08-aos625`. URL: `http://dx.doi.org/10.1214/08-AOS625`.