Kernel Search for Index Tracking with temporally compressed data

ashish1610dhiman@gmail.com

1 Abstract

The following work details the experiments conducted on various formulations of Enhanced Index Tracking problems. Precisely the research work is focused on the study of uni-period enhanced index tracking problem with a limited number of assets held in the tracking portfolio. Two different approaches to the problem are studied, namely Non-Linear Formulation and Linear Formulation. The Non-linear formulation is framed in Quadratic form and is solved using KKT conditions resulting in an analytical solution. While the linear formulation is solved through heuristic approach (more specifically Kernel Search algorithm). A combined approach is then formulated using the qualities of both the formulations referred above. Additionally, the application of dimension reduction techniques (namely NPCA and NMF) in the context of Enhanced Index Tracking is also analyzed to so as to replicate the index only at a macro level and thereby ignoring the minute (and futile) fluctuations. This essentially translates to limiting the resolution for tracking of benchmark to user decided period along with added benefit of decrease in computational complexity of the original problem. All the results are validated on both in sample and out of time data.

2 Introduction

Since the classical mean-variance model proposed by [5] the asset allocation problem has remained pertinent in the financial world and has been widely studied in the financial literature. Recent studies on the asset allocation problem (or Portfolio Optimisation) have extended it to cover several situations such as robust portfolio optimisation, multi period models etc. [1] .

One such class of Portfolio Optimisation problems is Index Tracking (or Enhanced Index Tracking). The index tracking actually caters to passive style of investing whereby the investor has to confirm to some predefined criteria. A common choice of criterion is reproducing the performance of market. The Index tracking problem thus essentially deals with determining a portfolio of assets (henceforth referred to as tracking portfolio) whose performance replicates, as closely as possible, that of a financial market index (or any arbitrary benchmark chosen). This style of investing thus helps investors track market's performance closely, while minimising overhead costs and reducing exposure to relatively illiquid assets. To describe the degree to which a Tracking portfolio is able to replicate the chosen benchmark's performance, a measure Tracking Error is introduced. It is a measure of the spread between the benchmark and the tracking portfolio.

Enhanced index tracking improves upon the original problem of Index Tracking by considering the excess return of the tracking portfolio (over the benchmark) along with tracking error. Essentially in Enhanced Index Tracking, the optimal portfolio is expected to outperform the benchmark with minimal additional risk over the index. Thus Enhanced Index tracking deals with two competing objectives i.e. the expected excess return of the portfolio over the benchmark and the tracking error from the benchmark. Enhanced index tracking is actually a recent investment strategy based on the idea of combining two different investment styles namely passive and active management. This is done in order to capture the strengths of both approaches.

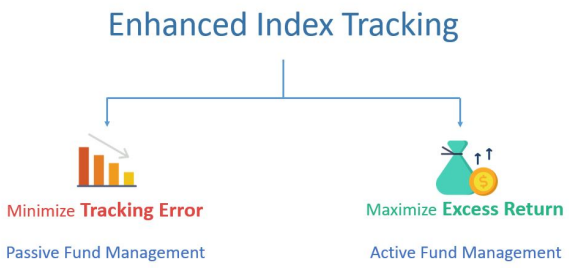

Figure 1: EIT: A combination of Active and Passive Management

Using a combination of 2 strategies enhanced index tracking aims to slightly outperform the benchmark

with minimal additional risk. Thus Enhanced Index Tracking is a type of portfolio optimization problem in which we seek to find an optimal allocation of securities so that the portfolio's return exceeds the return of a market index (benchmark) while also at the same time trying to mimic the index. The problem may be modelled in a number of ways by defining the objective function and the different constraints accordingly. A general form of which is:

$$\begin{aligned} &\underset{for \quad x \in feasible\ sol'n\ set}{\text{Minimize or Maximize}} && ObjectiveFunction(x) \\ &\text{subject to} && g_i(x) \leq b_i,\ i = 1, \ldots, m. \\ & && h_j(x) \leq 0,\ j = 1, \ldots, k. \end{aligned} \tag{1}$$

In the following sections we detail two such class of formulations. Namely *Quadratic Formulation* and *Linear Formulation*. Next we examine the role which dimension reduction could play in Enhanced Index Tracking Problems with results to support the same. Finally we propose a novel approach which is Linear (and Heuristic) in nature but leveraging the merits of both the classes. For experiments we consider financial markets where n stocks are available for the investment and assume that we observe the financial market over time periods t = 0, 1, - - - T. The models then determine an optimal portfolio composition at time T, to be held for time periods following T which we refer to as OOT. To this end we utilize a classical look-back approach, in which the optimal portfolio is determined using the data observed historically. The underlying assumption being that historical prices are good predictors for the future.

Important Terminologies

The terms extensively used in the following text are listed below:

- Firstly the index the model tracks is termed as **Benchmark**.
- The set of assets that is chosen to track a *Benchmark* is referred as the **Tracking portfolio**.
- The difference in return of the *Tracking portfolio* over the *benchmark* is called as **Excess Return**.
- Similarly **Tracking error** is a measure of the spread between the *Benchmark* and the *Tracking portfolio*.

3 Non Linear (Quadratic) Formulation of EIT

As discussed previously the enhanced index tracking problem (or enhanced indexation) aims at replicating a market index (or benchmark portfolio) as well as outperform the index by generating an excess return (over and above the return of the index), without purchasing all of the assets that make up the index. This structure has previously been studied by [1] and results in an analytical solution. This particular formulation is detailed in this section along with results of experiments conducted on the same.

3.1 Mathematical Model

First of all consider a Benchmark index with return P composed of n assets with returns $R_1, R_2, ...R_n$ and a covariance matrix $\Sigma > 0$ (i.e *a positive semi-definite matrix*). We denote by ω_i the weight invested in individual assets with return R_i , so that:

$$P = \sum_{i=1}^{n} \omega_i R_i \quad and \quad \sum_{i=1}^{n} \omega_i = 1$$

Note short positions (i.e. $\omega_i < 0$) are allowed for above formulation. Also consider a benchmark composed of n assets with weights as $\omega_B = [\omega_{B1}, \omega_{B2}, \omega_{B3}...\omega_{Bn}]^T$ and whose return $P_B = \sum_{i=1}^{n} \omega_{Bi} R_i$. Now the difference in the return of Tracking Portfolio and Benchmark can be expressed as:

$$P_e = P - P_B = (\omega - \omega_B)^T . R \tag{2}$$

where ω is vector of weights for Tracking Portfolio and $R = (R_1, R_2...R_n)^T$ is vector of returns of n assets. Note that the Returns here could be daily, weekly or any arbitrary level. Now the main identifiers for Enhanced Index Tracking Problem are *Excess Return (or μ_e) and Tracking Error (or σ_e^2)* which can be calculated using Expected value and Variance of P_e respectively.

$$\begin{aligned} \mu_e = E(P_e) &= E\left((\omega - \omega_B)^T . R\right) \\ &= (\omega - \omega_B)^T E(R) \end{aligned} \tag{3}$$

$$\begin{aligned}\sigma_e^2 = Var(P_e) &= E\left((P_e - \mu_e)^2\right) \\ &= (\omega - \omega_B)^T \Sigma (\omega - \omega_B)^T\end{aligned} \tag{4}$$

For the above tracking portfolio $|\omega| = |\omega_B| = n$ i.e. the Benchmark and Tracking Portfolio are composed of same n number of assets. If we were to impose a restriction on number of assets in Tracking Portfolio, such that Tracking Portfolio is composed of $p < n$ assets. The above constructs can be modified without loss of generality by selecting p assets randomly out of n which would now make up the Tracking Portfolio. Now the vector of weights for Tracking portfolio becomes $w = (\omega_1, \omega_2, ..., \omega_n)^T$ and the corresponding return vector becomes $\mathcal{R} = (R_1, R_2, ..., R_n)^T$ with covariance matrix ,$\Gamma > 0$ and $r = E(\mathcal{R})$. Next we define the covariance matrix between p and n assets as $\tilde{\Sigma}$ as $E\left((\mathcal{R} - r)(R - E(R))^T\right)$ Hence $P_e = w^T r - \omega_B^T R$. Plugging this into equation 2 and 3 we have:

$$\begin{aligned}\mu_e = E(P_e) &= E\left(w^T r - \omega_B^T R\right) \\ &= (w^T r - \mu_B)\end{aligned} \tag{5}$$

$$\begin{aligned}\sigma_e^2 = Var(P_e) &= E\left((P_e - \mu_e)^2\right) \\ &= w^T \Gamma w - 2w^T \tilde{\Sigma} \omega_B + \sigma_B^2\end{aligned} \tag{6}$$

where μ_B, σ_B^2 denote the Expected value and Variance of benchmark's return.

[2] considers the idea of single objective function made of weighted Tracking Error and Excess Return for Index Tracking and [1] extends this idea to Enhanced Index Tracking and proposes objective of the form $k_{tracking} * \sigma_e^2 - k_{return} * \mu_e$. Hence we borrow the below formulation from [1]:

$$\begin{aligned}\underset{w \in R^p}{\text{Minimize}} \quad & k_{tracking}\left(w^T \Gamma w - 2w^T \tilde{\Sigma} \omega_B + \sigma_B^2\right) - k_{return}\left(w^T r - \mu_B\right) \\ \text{subject to} \quad & w^T e = 1\end{aligned} \tag{7}$$

with $e = (1, 1, ..., p - times)^T$. Now the above model 6 is a Non Linear Optimisation problem. This can be solved by applying KKT conditions i.e finding a Lagrangian of the problem and setting its partial derivatives to 0. [1] solves the problem by said method and the reader can refer to it to see calculation of analytical solution. This work however directly presents the optimal solution:

$$w^* = \Gamma^{-1}\left(\tilde{\Sigma}\omega_B + \frac{k_{return}}{2k_{tracking}} r + \frac{\tau}{\alpha} e\right) \tag{8}$$

where $\tau = 1 - e^T \Gamma^{-1} \tilde{\Sigma} \omega_B - k_{return} \frac{e^T \Gamma^{-1} r}{2k_{tracking}}$ and $\alpha = e^T \Gamma^{-1} e$

Next similar to [1] we compare the performance of optimal tracking portfolio to one without tracking (a derivation of Markowitz's model), i.e. solution of:

$$\begin{aligned}\underset{w \in R^p}{\text{Minimize}} \quad & k_{tracking}(w^T \Gamma w) - k_{return}(w^T r) \\ \text{subject to} \quad & w^T e = 1\end{aligned} \tag{9}$$

The solution of this model is given by:

$$\tilde{w} = \Gamma^{-1}\left(\frac{k_{return}}{2k_{tracking}} r + \frac{\tilde{\tau}}{\alpha} e\right) \tag{10}$$

where $\tilde{\tau} = 1 - k_{return} \frac{e^T \Gamma^{-1} r}{2k_{tracking}}$ and $\alpha = e^T \Gamma^{-1} e$

Note as visible in 8 and 10 and noted by [1] w^* and $\tilde{w}$ is only dependent on ratio $\frac{k_{return}}{k_{tracking}}$. This is further validated in 3.2 section below, which details results of experiments conducted on this formulation. It is also to be noted that $(k_{tracking}, k_{return}, p)$ is common a set of parameters for both the models.

3.2 Results and Remarks

The data considered for the following experiments was taken from NSE website. It consists of daily price data of 20 stocks listed on NSE across a period of 2 years starting from August 2016 to August 2018. We have solved model 7 and 9 on this data using python. Also since the data did not contain an benchmark index explicitly. So to counter this we create a dummy benchmark index by solving the without Tracking problem (i.e. 9 but with p=n. This is detailed in the graphic below:

Following the above framework experiments are conducted. Our data had 500 days of data. We use first 350 days to solve the problem and next 150 days for Out of Time (OOT) validation. Below are results for same:

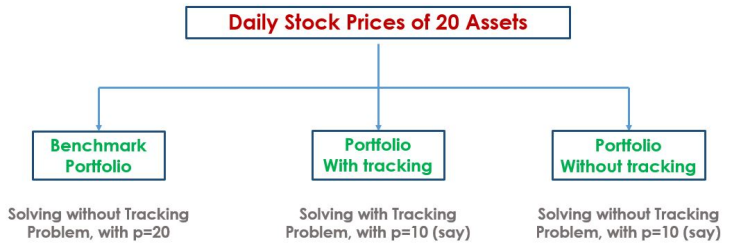

Figure 2: Non Linear formulation experiments

Model Parameters				Avg Portfolio daily return (OOT)			Performance (OOT)	
p	*tracking/return*	*k_tracking*	*k_return*	*benchmark*	*without_tracking*	*with_tracking*	*excess_return*	*tracking_error*
	1.5	0.9	0.600	0.027693	0.003165	0.005731	-0.021962	0.007808
6	1.5	1	0.667	0.027693	0.003165	0.005731	-0.021962	0.007808
	1.5	1.1	0.733	0.027693	0.003165	0.005731	-0.021962	0.007808
	1.5	0.9	0.600	0.027693	0.003166	0.005748	-0.021944	0.007828
7	1.5	1	0.667	0.027693	0.003166	0.005748	-0.021944	0.007828
	1.5	1.1	0.733	0.027693	0.003166	0.005748	-0.021944	0.007828
	1.5	0.9	0.600	0.027693	0.003142	0.005711	-0.021982	0.007572
8	1.5	1	0.667	0.027693	0.003142	0.005711	-0.021982	0.007572
	1.5	1.1	0.733	0.027693	0.003142	0.005711	-0.021982	0.007572

Table 1: Impact of $\frac{k_{return}}{k_{tracking}}$ on Performance

1. The solution of both problems (i.e. with and without tracking) depends only on ratio of $\frac{k_{return}}{k_{tracking}}$ and choice of p. Both $|p|$ and the actual 'p' stocks chosen impact performance.
2. By an appropriate choice of the ratio $\frac{k_{return}}{k_{tracking}}$, a manager could choose one of the three investment strategies: (i) achieve an average return higher than a reference index (active management), (ii) track a reference index with a positive excess return (enhanced index tracking) or (iii) replicate the return of a reference index (index tracking).[1]

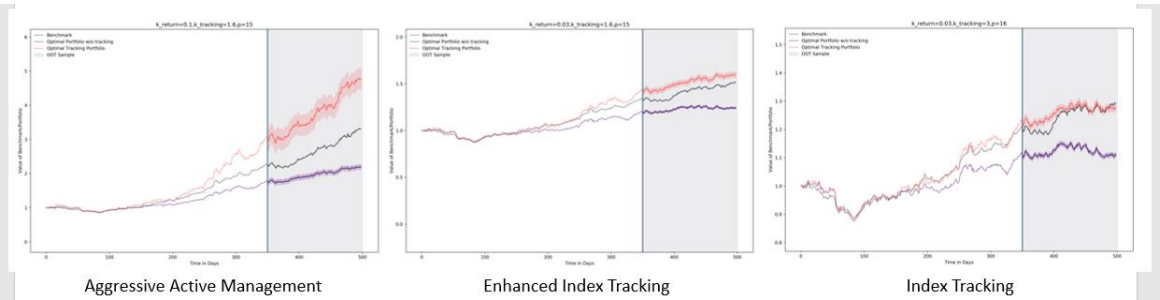

Figure 3: Three Styles of Portfolio Management

3. The input to tracking model is ω_B or the weights of the benchmark. Although [1] suggests an approach to modify Excess return and Tracking Error as:$\mu_e = w^T r - E(R_B)$ and $\sigma_e^2 = w^T \Gamma w - 2\sigma_B^2 w^T \beta + \sigma_B^2$ where R_B is return of benchmark σ_B^2 is market variance, and β is vector of betas of chosen p assets. An example of ω_B being unavailable is a Fund of Funds. In such cases, betas of individual stocks might not always be readily available especially for relatively illiquid stocks, thereby weakening the premise of Enhanced Index Tracking.

4 Linear Formulation of EIT

A Mixed Integer Linear Problem (MILP) formulation of the Enhanced Index Tracking problem has been detailed in this section. The core structure of this formulation has been borrowed from [3] which deals with both the Bi-Objective and Mono-objective MILP formulation of EIT. Now since Integer Programming is an NP-complete problem, even solving relatively small problems might become hard. Many techniques like Branch and Bound and Branch and Cut exist to solve Integer Programming Problems. Also Branch and Bound can be modified by using pruned problem to accommodate heuristics. But for our use case we employ the Kernel Search algorithm, similar to [3] , to arrive at a solution of the problem. Kernel Search Algorithm as an heuristic framework to solve Index Tracking is detailed in the work [4].

4.1 Mathematical Model

We consider an investor is holding a portfolio of stocks, hereafter referred to as the *current portfolio*. We aim to rebalance the current portfolio composition in time period T so as to maximize the portfolio average excess return over the benchmark (I) while simultaneously minimizing the tracking error. Also in this formulation, real-world constraints which were absent in the erstwhile discussed Non-Linear formulation 3 such as transaction costs etc. are added. To define all the components of the model let us define the required variables as:

- T be the Time period under consideration (i.e. in sample time)
- $q_{j,t}$ be the price of j^{th} stock at time t
- I_t be the value of benchmark at time t]
- $r_{j,t} = \left(\frac{q_{j,t}-q_{j,t-1}}{q_{j,t-1}}\right)$ be the return of j^{th} stock at time t
- $r_{I,t} = \left(\frac{I_t-I_{t-1}}{I_{t-1}}\right)$ be the return of benchmark at time t
- X_j^0 be the units of j_{th} stock in current portfolio
- X_j^0 be the units of j_{th} stock in rebalanced portfolio
- C is the total capital sum available to investor for investing
- τ is the spare cash available to investor for investing over value of current portfolio

Hence it follows,

$$C = \sum_{j=1}^{n} q_(j,t)X_j^0 + \tau$$

In this model, X_j^1 are the core decision variables. Also contrary 3 we impose constraints $X_J^0 > 0$ and X_j^1 here, thereby disallowing Short Selling. Also while rebalancing the current portfolio, it is only natural that an investor can only sell as much he owns i.e. if $X_j^1 \leq X_j^0$ then $(X_j^0 - X_j^1) \in [0, X_j^1]$. And if he buys that stock then $X_j^1 > X_j^0$. In both the scenarios we assume there is a transaction cost to be borne by the investor, proportional to value of transaction i.e $q_{j,T}|X_j^1 - X_j^0$. For generality we assume transaction cost varies with stock and whether investor is buying or selling the stock. Over the cost proportional to value of transaction, we assume there is also a fixed transaction cost levied on investor irrespective of if he's selling or buying the stock. This total transaction cost is constrained to be a fraction of Capital available. Imposed also is lower and upper bound on capital to be invested in a single stock, to prevent high exposure to one asset. Finally, a cardinality constraint is imposed on size of rebalanced portfolio. The required variables to model the above constraints are:

- c_j^b is proportionality constant for buying cost of j_{th} stock
- c_j^s is proportionality constant for buying cost of j_{th} stock
- b_j is the buying value of j_{th} stock
- s_j is the selling value of j_{th} stock
- f_j is the selling value of j_{th} stock
- w_j is a binary variable which takes value 1 if j_{th} stock is sold
- ρ is the Max Transaction Cost fraction
- λ_j is the lower bound constant for investment in j_{th} stock
- ν_j is the lower bound constant for investment in j_{th} stock
- y_j is a binary variable which takes value 1 if j_{th} stock is present in rebalanced portfolio (i.e $X_j^0 > 0$)
- k is the cardinality limit on rebalanced portfolio
- ξ is the Tracking Error proportionality constant

Hence we define the excess return (z_1) of tracking portfolio over benchmark as the absolute excess value of portfolio over benchmark averaged over time, i.e:

$$z_1 = \frac{1}{T}\sum_{t=1}^{T}\left[\sum_{j=1}^{n} r_{j,t}q_{j,T}X_j^1 - r_{I,t}C\right]$$

Tracking Error (TrE) is defined as absolute deviation of portfolio from benchmark averaged over time. Note the linear nature of Tracking Error here compared to quadratic in 3.

$$TrE = \sum_{t=1}^{T}\left|\theta I_t - \sum_{j=1}^{n} q_{j,t}X_j^1\right|$$

where $\theta = \frac{C}{I_T}$ is used to scale the value of Benchmark. Let d_t and u_t be the variables depicting downside and upside deviation of tracking portfolio from benchmark at time t. Hence it follows, $d_t - u_t = \theta I_t - \sum_{j=1}^{n} q_{j,t}X_j^1$ for $t = 1, 2, \cdots, T$. Thus Tracking Error can be expressed as:

$$TrE = \sum_{t=1}^{T}(d_t + u_t)$$

Hence in this model each stock is represented by two binary variables (y_j and w_j) and three non-negative continuous variables (X_j^1, b_j, s_j). Thus from scope of the model each stock can be represented as a vector § composed of the decision variables.

The exact optimisation problem, henceforth referred to as $\mathcal{EIT}$ can be summed up as:

$$\begin{aligned}
\underset{x\ \in\ \mathcal{X}}{\text{Maximize}} \quad & z_1 = \frac{1}{T}\sum_{t=1}^{T}\left[\sum_{j=1}^{n} r_{j,t}q_{j,T}X_j^1 - r_{I,t}C\right] \\
\text{subject to} \quad & \sum_{t=1}^{T}(d_t + u_t) \leq \xi C \\
& d_t - u_t = \left(\theta I_t - \sum_{j=1}^{n} q_{j,t}X_j^1\right) \forall t = 1, 2, \cdots, T \\
& \lambda_j C y_j \leq X_j^1 q_{j,t} \leq \nu_j C y_j \ \forall j = 1, 2, \cdots, n \\
& \sum_{j=1}^{n} y_j \leq k \\
& \sum_{j=1}^{n} X_j^1 q_{j,t} = C \\
& b_j - s_j = (X_j^1 - X_j^0)q_{j,t} \ \forall j = 1, 2, \cdots, n \\
& b_j \leq \left(\nu_j C - X_j^0 q_{j,T}\right) \ \forall j = 1, 2, \cdots, n \\
& s_j \leq X_j^0 q_{j,T} w_j \ \forall j = 1, 2, \cdots, n \\
& \sum_{j=1}^{n}\left(c_j^b b_j + c_j^s s_j + f_j w_j\right) \leq \rho C \\
& X_j^1, b_j, s_j \geq 0 \ \forall j = 1, 2, \cdots, n \\
& d_t, u_t \geq 0 \ \forall t = 1, 2, \cdots, T \\
& y_j, w_j \in \{0, 1\} \ \forall j = 1, 2, \cdots, n \\
& if\ X_j^0 = 0\ then\ y_j = w_j \ \forall j = 1, 2, \cdots, n
\end{aligned} \tag{11}$$

where C, τ, λ_j, ν_j, ρ, c_j^b, c_j^s, f_j, ξ are the different parameters of the model. The solution to this model leverages Kernel Search framework proposed by [3] in its skeletal form with some incremental changes.

4.1.1 Kernel Search

The basic idea of the Kernel Search is to intensively explore a sequence of relatively small/moderate-size portions of the solution space. This helps to abstract Kernel as a set of securities. The exploration in space

is guided by the identification of subsets of decision variables and the subsequent solution of the resulting restricted problems by means of a general-purpose MILP solver [3]. We further follow the notation of $\mathcal{EIT}(\mathcal{N})$ as EIT model 11 being solved for N stocks, $\mathcal{LP}(N)$ as Linear Relaxation of $\mathcal{EIT}(\mathcal{N})$ and $\mathcal{EIT}(\mathcal{K})$ as EIT model solved on Kernel, where $\mathcal{K}$ is a subset of stocks. The Kernel Search Framework can be broken into two stages. The first stage is *Initialisation Phase* where $\mathcal{LP}(N)$ is solved to intelligently rank the promising stocks in a list $\mathcal{L}$ and also to get the best possible value of $z1$. The initial kernel is chosen out of the first m stocks of $\mathcal{L}$, while the rest $(n-m)$ stocks are partitioned into N_b disjoint buckets. Next $\mathcal{EIT}(\mathcal{K})$ is solved to give a lower bound of $z1$. The second stage is iterative over $h = 1, 2 \cdots N_b$ buckets , and is referred to as *Solution Phase*. At each iteration a restricted problem, $\mathcal{EIT}^*(\mathcal{K} \cup bucket_h)$ is solved, where $\mathcal{EIT}^*$ is $\mathcal{EIT}$ but with two additional constraints, namely:

1. Optimal value of previous feasible iteration is used as a lower threshold for current iteration,
2. At least one stock should be selected from current bucket B_h i.e $\sum_{j \in bucket_h}(z_j) \geq 1$

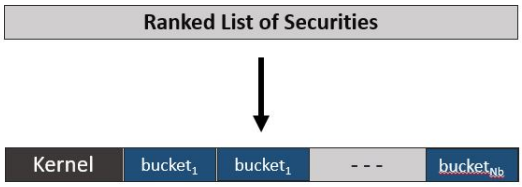

Figure 4: Sorting and Bucketing the Stocks

The exact Kernel Search Framework is described below:

Algorithm 1: Kernel Search Framework for solution of $\mathcal{EIT}(\mathcal{N})$

Data: Set of $\mathcal{N}$ securities
Result: optimal solution (x^*, z^*); or *failure*=**True**
Step 1: *Initialisation Phase*;

1. Set *failure* = **False**
2. If $\mathcal{LP}(N)$ can be solved feasibly, store its solution as (x_{lp}, z_{lp}), else set *failure* = **True** and **STOP**
3. Sort the set $\mathcal{N}$ as per a predefined criterion leveraging solution of $\mathcal{LP}(N)$ and create the ranked list $\mathcal{L}$
4. Using this ranked list $\mathcal{L}$:
   - build a initial kernel $\mathcal{K}$ by taking first **m** members of list $\mathcal{L}$:,
   - partition the remaining securities into $\{bucket_h\}_{h=1,2\cdots N_b}$ buckets as per previously decided sequence
5. Solve sub-problem $\mathcal{EIT}(\mathcal{K})$. If feasible solution exists let it be (x^*, z^*) , otherwise set *failure*=**True**

Step 2: *Solution Phase*;
for h =1 **to** N_B **do**
  Set $\mathcal{K}' = (\mathcal{K} \cup bucket_h)$
  if $\mathcal{EIT}^*(K')$ *has feasible solution* (x', z') **then**
    if *failure* = ***True*** **then** set *failure* = **False**;
    Set $(x^*, z^*) \leftarrow (x', z')$
    `// Let` K_h^+ `be securities belonging to` $bucket_h$ `selected in solution (x',z')`
    `// Let` K_h^- `be securities belonging to Kernel` $\mathcal{K}$ `that have not been selected since last p iterations in solution (x',z')`
    Set $\mathcal{K}' = \left(\mathcal{K} \cup K_h^+\right) - K_h^-$
  end
end

The Kernel Search algorithm thus adds two other user defined parameters, namely m = kernel size and l_{buck} = the length of bucket, to the model. Using the Kernel search algorithm, we solve the EIT model framed in 11. The following section details the results of experiments conducted on it.

4.2 Results and Remarks

The data used for these experiments is derived from the publicly available OR-Library. It consists of 8 benchmark instances, namely Hang Seng (Hong Kong), DAX100 (Germany), FTSE100 (United Kingdom), S&P100 (USA), Nikkei225 (Japan), S&P500 (USA), Russell2000 (USA) and Russell3000 (USA). The data provides 291 weekly prices from March 1992 to September 1997 for the benchmark and its component securities. The number of securities varies from 31 in Hang Seng to 2151 in Russell3000, thereby allowing us to validate results in higher dimensional space.

We consider the first 200 weekly prices as in sample, to find optimal performance, and then test the OOS performance on the remaining 91 weeks. For the sake of simplicity we assume $c_j^b = c_j^s = 0.01 \ \forall$ stocks j, which essentially makes buying/selling costs proportion agnostic of stocks.

			Average					Average	
k	ρ	**z_value**	**z_linear**	**#steps**	k	m	**z_value**	**z_linear**	**#steps**
	0.1	1633.193709	1732.751987	0.666667		8	1633.193709	1732.751987	2.000000
10	0.2	1633.193709	1732.751987	0.666667	10	12	1633.193709	1732.751987	0.000000
	0.5	1633.193709	1732.751987	0.666667		16	1633.193709	1732.751987	0.000000
	0.1	1729.275617	1732.751987	2.666667		8	1729.275617	1732.751987	4.666667
16	0.2	1729.275617	1732.751987	2.666667	16	12	1729.275617	1732.751987	2.666667
	0.5	1729.275617	1732.751987	2.666667		16	1729.275617	1732.751987	0.666667
	0.1	1732.751987	1732.751987	4.000000		8	1732.751987	1732.751987	6.000000
25	0.2	1732.751987	1732.751987	4.000000	25	12	1732.751987	1732.751987	4.000000
	0.5	1732.751987	1732.751987	4.000000		16	1732.751987	1732.751987	2.000000

Table 2: Effect of k & ρ and k & m on solution of $\mathcal{EIT}(\mathcal{N})$

As expected the optimum value of objective function increases with higher k .We see choice of ρ does not impact neither the solution nor the steps taken to reach optimum solution. While with a lower m, we are able to reach the same solution in quicker steps.

The two most important parameters impacting performance of $\mathcal{EIT}(\mathcal{N})$ are k and ξ, with increasing k, both in-sample and oos excess return increase, while choice of ξ impacts Tracking Performance.

k	ξ	z_{in_sample}	TrE_{in_sample}	z_{oos}	TrE_{oos}	**#steps**
10	1.2	1530.430358	224172.993279	3613.340702	-1.520349e+07	0.666667
	1.3	1637.074329	216741.760970	8291.844229	-2.024084e+07	0.666667
	1.4	1732.076441	261853.287566	8918.196357	-2.093902e+07	0.666667
16	1.2	1648.811498	85576.224460	6463.858074	-3.787245e+07	3.000000
	1.3	1729.443446	134887.449248	6857.818146	-3.817990e+07	3.000000
	1.4	1809.571907	126906.369317	7365.981180	-2.936720e+07	2.000000
25	1.2	1653.519570	106142.643581	7406.139759	-3.902740e+07	4.000000
	1.3	1733.535995	102298.539670	7258.858578	-3.885810e+07	4.000000
	1.4	1811.200395	147940.860604	7850.030115	-3.965217e+07	4.000000

Table 3: Effect of k & ξ on in-sample & OOS (out of sample) performance of $\mathcal{EIT}(\mathcal{N})$

5 Dimensionality Reduction across time

While tracking a benchmark in long term, we hypothesize it becomes insignificant to track the daily nuances of the benchmark. Rather the macro trends over bi-weekly/weekly periods are far more critical in such tracking requirements. This argument is supported by research [6] which shows Dimension reduction of prices in the Markowitz Mean-variance model directly tackles the complex problem of large-scale portfolio construction. We borrow this approach and apply it to our linear model. This research shows that dimension reduction for portfolio optimization can improve the overall portfolio performance by providing "better risk-return characteristics". Apart from the better performance characteristics, Dimension reduction also significantly reduces the computational complexity of the problem.

The following section details a comparative analysis and application of two fundamental non-negative dimensionality reduction methods, namely the non-negative matrix factorization (NMF) and non-negative principal components analysis (NPCA), for application in Enhanced Index Tracking. The study is limited specifically to Non-Negative method since negative stock prices have an intangible meaning.

As the number of observations increase, oscillations between their returns increase and thus reductions start producing outliers. To avoid these outliers, the first step of the dimension reduction methodology requires dividing the unreduced dataset into k equidistant time windows X_{ti} as described in 5.

$$
\begin{aligned}
Unreduced\ Data &= [StockPrices]_{T\ *\ n} \\
X_{ti} &= [StockPrices]_{(T/k)\ *\ n} \\
i.e. Unreduced\ Data &= [X_{t1}, X_{t2}, - - -, X_{tk}]^T
\end{aligned}
\tag{12}
$$

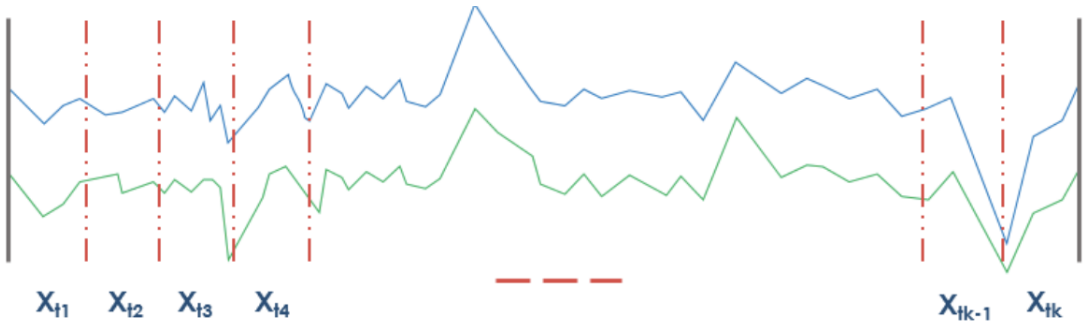

Figure 5: Dividing original time series to equal size windows

Next Dimensionality Reduction is applied to decrease the time-dimension of each of these X_{ti}. The reduced dimensions for each of the X_{ti} are then combined into one on the basis of SVP (Statistical Variance Procedure) which essentially takes a weighted sum of the reduced dimensions with weights being the proportion of variance explained in the original data. Hence we derive a single vector f_i of stock prices for each of the k subsets of data. These reduced prices then become the input (Reduced Data=$[f_i]_{k*n}$) for our tracking problem.

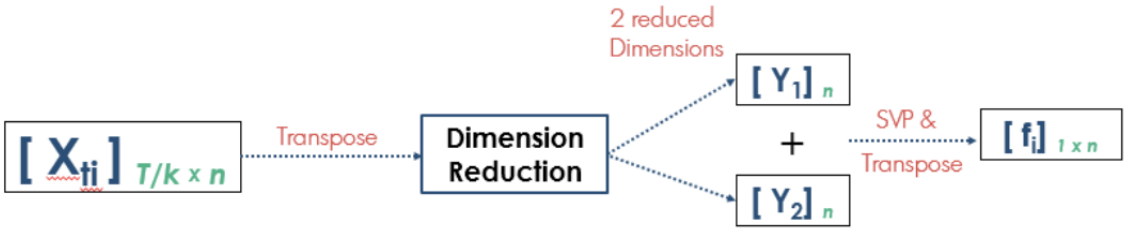

Figure 6: Dimension Reduction Framework

The following sections briefly describe the NPCA and NMF methods used for this exercise and the decision criteria for choosing the number of reduced dimensions.

5.1 NPCA

PCA is a widely used unsupervised dimensionality reduction method that computes the linear combinations of the variables of a dataset to represent it through a new set of linearly uncorrelated and orthogonal variables, namely, the principal components. In PCA, the first principal component is always the component representing the largest variance. The second principal component is uncorrelated with the first one and its variance is the second largest among all the variances of all the components, and this pattern follows. PCA maintains the maximum variance of the original dataset by representing its compressed version at a lower dimensional space.

$$
\begin{aligned}
\text{Maximize} \quad & v^T Q v \\
\text{vTv=1} \quad & v \geq 0 \\
& where\ Q = Cov(X)
\end{aligned}
\tag{13}
$$

The solution to this model is the normalized eigen-vector corresponding to the largest eigenvalue and the first principal component of Q. Where v is a coefficient vector v, Q is covariance matrix of X and if d be an arbitrary linear combination of original data [X].

5.2 NMF

NMF is a dimension reduction method for extracting features from non-negative data. NMF decomposes the original non-negative data matrix $[X]_{n*p}$ into two matrices; the basis matrix $[B]$, and the mixing coefficients

matrix $[H]$ that indicates the weights of the elements in [B]. NMF tries to approximate:

$$[X]_{n*p} \approx [B]_{n*h} \times [H]_{h*p} \tag{14}$$

Determining the factorization rank is a central issue in NMF since the information in [X] is transferred to r number of factors in the columns of [B]. Gillis (2014) reference 4 states, the factorization rank of PCA can be used to determine the factorization rank of NMF. As for the NPCA model in Eq. (9) requires non-negative input and to avoid negative prices, the symmetric input matrix that computes the non-negative principal components were not centred the column means were not subtracted from the observations.

6 Combined Approach

Both the formulation of EIT which are discussed above have their own pros and cons. The motivation for this approach is to use the best of both approaches and formulate a novel approach for formulating Enhanced Index Tracking. Quadratic Formulation gives us the flexibility to alter the importance of return and tracking error as per the risk aversion of the user. Also, it suffers from almost no feasibility issues. But this formulation is too simple and does not include real-world characteristics such as Transaction Costs, No Short-Selling etc. Also, the model requires us to input the weights of the benchmark which are sometimes inaccessible.

While on the other hand linear formulation does model the real world characteristics such as Transaction costs etc. And also does not require the weights of underlying stocks in the index but only the price of the index itself. But this formulation is seriously plagued by the feasibility issues which are chiefly created due to Tracking Error.

Hence a midway formulation is created by borrowing features from both the formulations. Concretely all the constraint of the linear formulation are taken but the Tracking Error Constraint which is rather included in the objective. We also introduce separate weights for downside and upside tracking errors, to add to the flexibility of the model.

WIP below this:

6.1 Mathematical Model

Need to update

As discussed we borrow majority of the framework from the Linear model, albeit with some slight changes as given below 11. The exact optimisation problem, henceforth referred to as $\mathcal{EIT}_{\lceil}{}^{\sqcap\dashv\updownarrow}$ can be represented as:

$$\underset{x \in \mathcal{X}}{\text{Maximize}} \quad z = w_{return} \quad z_1 = \frac{1}{T}\sum_{t=1}^{T}\left[\sum_{j=1}^{n} r_{j,t}q_{j,T}X_j^1\right.$$

$$z_2 \sum_{t=1}^{T}(w_{down} * d_t + w_{up} * u_t) \leq \xi C$$

$$\text{subject to} \quad \begin{aligned}
&\sum_{t=1}^{T}(d_t + u_t) \leq \xi C \\
&d_t - u_t = \left(\theta I_t - \sum_{j=1}^{n} q_{j,t}X_j^1\right) \forall t = 1, 2, \cdots, T \\
&\lambda_j C y_j \leq X_j^1 q_{j,t} \leq \nu_j C y_j \ \forall j = 1, 2, \cdots, n \\
&\sum_{j=1}^{n} y_j \leq k \\
&\sum_{j=1}^{n} X_j^1 q_{j,t} = C \\
&b_j - s_j = (X_j^1 - X_j^0)q_{j,t} \ \forall j = 1, 2, \cdots, n \\
&b_j \leq \left(\nu_j C - X_j^0 q_{j,T}\right) \ \forall j = 1, 2, \cdots, n \\
&s_j \leq X_j^0 q_{j,T} w_j \ \forall j = 1, 2, \cdots, n \\
&\sum_{j=1}^{n}\left(c_j^b b_j + c_j^s s_j + f_j w_j\right) \leq \rho C \\
&X_j^1, b_j, s_j \geq 0 \ \forall j = 1, 2, \cdots, n \\
&d_t, u_t \geq 0 \ \forall t = 1, 2, \cdots, T \\
&y_j, w_j \in \{0, 1\} \ \forall j = 1, 2, \cdots, n \\
&if \ X_j^0 = 0 \ then \ y_j = w_j \ \forall j = 1, 2, \cdots, n
\end{aligned} \tag{15}$$

where C, τ, λ_j, ν_j, ρ, c_j^b, c_j^s, f_j, ξ are the different parameters of the model. The solution to this model leverages Kernel Search framework proposed by [3] in its skeletal form with some incremental changes.

The 2 core propositions of the paper are:

1. New Enhanced Index Tracking Approach (EIT_{dual})
2. Benefits of Reduced Data on the two approaches EIT_{basic} and EIT_{dual}
   (a) Higher Slope of *return_pu_risk* vs k, i.e. with increase in k, higher increase in *return_pu_risk*
   (b) Subdued Slope of *return_pu_risk* vs ρ, i.e. with increase in ρ, lower decrease in *return_pu_risk*

To prove the above propositions we use the quantitative approach. To this end we use 2 data sources,

1. Hang Seng:
   31 stocks — March 1992 to September 1997 — Gradual up trajectory of market — weekly, NPCA reduced, NMF reduced
2. S&P500:
   500 stocks — Feb 2013 to Mar 2018 — Explosive up trajectory, following static market — daily, weekly, NPCA reduced, NMF reduced

7 Results

7.1 Qualifying Metrics

7.1.1 Return per unit risk (*return_pu_risk*)

We define *return_pu_risk* as:

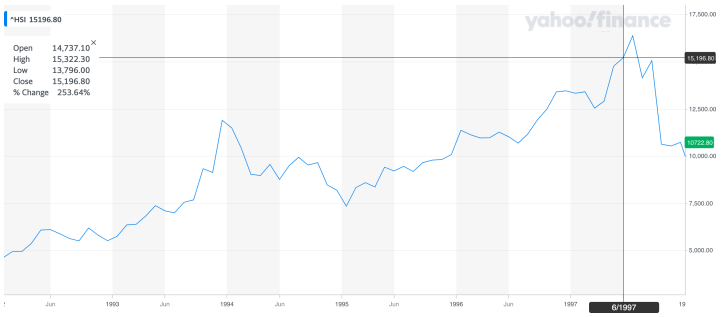

Figure 7: Hang Seng Trend

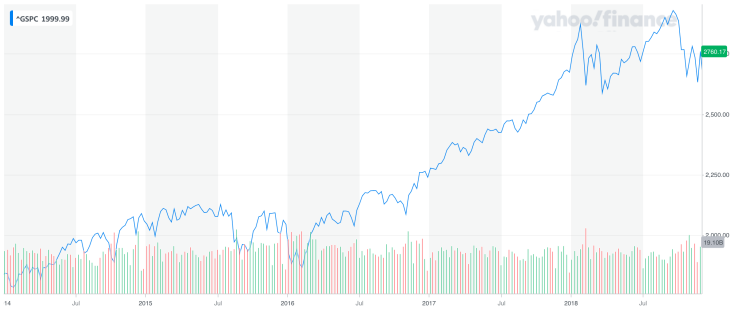

Figure 8: S&P500 Trend

$$return\_pu\_risk = \frac{ExcessReturn}{TrackingError}, where:$$
$$ExcessReturn = \frac{\sum^{T}(portfolio_{return} - index_{return})_t}{T}$$
$$TrackingError = \frac{\sum^{T}|portfolio_{value} - index_{value}|_t}{T}$$

Note since $ExcessReturn$ and $TrackingError$ are not necessarily normalised,the metric becomes a little tricky to be compared across data sets with different scaling (for ex: Hang Seng weekly against NPCA bi-weekly).

Hence to mitigate this, we take the solution (i.e portfolio weights) as from reduced data sets, and transpose them to the un-reduced dataset. This is detailed more in the section 7.4

7.2 NMF vs NPCA

It is seen that both NMF and NPCA follow the trend of the original time series, though NMF resembles the original data more closely but is also exhibits greater variance.

Also since both types of dimensionality reduction, alter the scale of stock and index prices. Hence for a better comparison, we scale the parameters, C:Capital & f: Fixed Txn cost on reduced data.

We find the scaling factor as follows:

$$scaling_{factor} = \frac{C_{unreduced}}{price_{index,T}}$$

7.3 Proposition 1

To prove EIT_{dual} is better than EIT we use the following steps:

1. Run EIT and EIT_{dual} under same set of common parameters for the above listed datasets.
2. From Excess return and Tracking Error, compute $return_pu_risk$as (Excess Return) / (Tracking Error). Both the approaches use consistent definitions of two.

		Hang Seng			*S&P500*			
		weekly	**NPCA**	**NMF**	**daily**	**weekly**	**NPCA**	**NMF**
T		*291*	*59*	*59*	*1098*	*237*	*297*	*297*
#param combinations		243	243	243				
Avg(EIT_dual >EIT) on param combinations	*In Sample (70%)*	0.67	0.67	0.0				
	OOS (30%)	0.93	0.49	0.93				

Table 4: EIT_basic vs EIT_dual for various settings

EIT_type	**dual**			**basic**		
Dataset — Hang Seng	*1*	*npca*	*nmf*	*1*	*npca*	*nmf*
return_pu_risk_sample_transpose	1.356284e-07	1.961903e-06	-7.874494e-07	1.962252e-05	3.507791e-05	6.306736e-07
return_pu_risk_oot_transpose	-4.498208e-07	1.359339e-06	2.753823e-06	1.032645e-04	2.727054e-05	-3.150138e-05

Table 5: Slope of return/risk vs k (16-25)

The below of table gives the % of instances where EIT_{dual} was better than EIT_{basic}. Note for many of the instance settings, EIT_{basic} yields unfeasible solution, which are not counted in the below numbers. (Adding them would further tilt the scale in favour of EIT_{dual})

Note looking at the left table one might think, that using reduced data does not help basic vs dual criteria. However we must remember Hang Seng's base version was already weekly and reducing it bi-weekly through either NPCA or NMF to a bi-weekly level, might be decreasing the effective training data (rather than intended reduction in noise). To prove this point we'd need to see results on the right hand side of table, where we potentially see reduced data helping Basic vs Dual more.

Also as discussed in 7.2 NPCA retains lesser variance of original time-series, and more closely following on the overall trend. This helps in increasing in sample performance on Hang Seng, however, it loses on OOT, b'cos of missing the sudden drop as seen in 7. Conversely NMF, does better on OOT.

7.4 Proposition 2

For hypothesis put under 7.3, we did not need to make comparisons between different time frames. However, the idea of this propositions warrants so.

To achieve so, we use the following methodology:

1. X_1 values from the solution of reduced problem (NPCA/NMF)
2. Convert X_1 values to weights w as $X_1 * price_{reduced}[t]/C_{reduced}$
3. Get $X_{1,new}$ as $\frac{w*C}{price_{unreduced}[t]}$
4. Use $X_{1,new}$ to calculate metric $return\_pu\_risk$

Note we still can't compare values across Hang Seng and S&P500 as they still are on different scales.
*Note this is over the scaling as mentioned in 7.2

7.4.1 Proposition 2(a)

We propose using reduced data has the effect of improving risk/return characteristics with increasing k.

From the below table, we see reduced by NPCA, helps both dual and basic approach, but more so for dual approach. This is in line with results of Proposition 1, i.e. since bi-weekly reduced data has lowest amount of information available, adding extra vars in form of securities best increases performance. However basic might already have significant information from weekly data, hence adding more securities to portfolio provides lesser marginal utility.

We also see that basic with NPCA has the highest slope. This again might be tied down to the fact, that the basic approach had lower performance than the dual approach, which consequently might give it a better scope of improving the results. While on the other hand, dual might already be reaching saturation in terms of information extracted from the current set of securities.

7.4.2 Proposition 2(b)

We propose using reduced data has the effect of sensitising risk/return characteristics to ρ.

This proposition does not hold.

EIT_type	dual			basic		
Dataset — Hang Seng	*1*	*npca*	*nmf*	*1*	*npca*	*nmf*
return_pu_risk_sample	5.535120e-03	-2.942339e-01	4.733874e-02	1.496724e-03	-4.325654e-02	7.698764e-03
return_pu_risk_oot	-6.281369e-03	-1.411890e-01	-6.572610e-02	-5.886736e-04	9.324724e-02	-9.357021e-03

Table 6: Slope of return/risk vs pho (0.2-0.4)

References

[1] Wanderlei Lima [de Paulo], Estela Mara [de Oliveira], and Oswaldo Luiz [do Valle Costa]. Enhanced index tracking optimal portfolio selection. *Finance Research Letters*, 16:93 – 102, 2016.

[2] Christian Dose and Silvano Cincotti. Clustering of financial time series with application to index and enhanced index tracking portfolio. *Physica A: Statistical Mechanics and its Applications*, 355(1):145 – 151, 2005. Market Dynamics and Quantitative Economics.

[3] C. Filippi, G. Guastaroba, and M.G. Speranza. A heuristic framework for the bi-objective enhanced index tracking problem. *Omega*, 65:122 – 137, 2016.

[4] G. Guastaroba and M.G. Speranza. Kernel search: An application to the index tracking problem. *European Journal of Operational Research*, 217(1):54 – 68, 2012.

[5] Harry Markowitz. Portfolio selection*. *The Journal of Finance*, 7(1):77–91, 1952.

[6] Halit Alper Tayalı and Seda Tolun. Dimension reduction in mean-variance portfolio optimization. *Expert Systems with Applications*, 92:161 – 169, 2018.