1.5.4. Moment Matching for scenario generation.

This method could be based on an optimization model or some other heuristic where the objective is to obtain a set of scenarios such that the resulting moments, usually the first four, are close or equal to the moments obtained from historical data or some other targets set by the decision maker (see Hoyland and Wallace (2001)).

A potential shortcoming is non-convexity since it may prevent finding a perfect match for the moments due to the high probability of obtaining a locally optimum solution. Even though the global optimum cannot be found, it can be still acceptable to have statistical features that are sufficiently good approximations to the target moments from a SP point of view. The nonlinearity in this approach can be eliminated by linearizing the objective function such that it is a weighted sum of absolute deviations from the target moments instead of squared deviations.

1.6.1. Scenario Generation Methodology

The scenario generation algorithm is based on the moment matching technique which aims to find a discrete probability distribution for the scenarios such that the first four moments of the generated scenarios are closely matched with the historical data. We consider a linear program and take security returns and interest rates as given parameters (generated before the matching).

In addition, we incorporate state dependency of variance via the EGARCH model, which, to the best of our knowledge, is included into a scenario tree construction for the first time. Instead of node-by-node optimization, we consider solving a single optimization model that optimizes the entire scenario tree at once.

1.6.2. Portfolio Optimization Model

Once the scenario tree is constructed, the next step is to input the scenario tree into an SP model to obtain the investment decisions. The advantage of SP models is the ability to consider various objective functions and constraints. Therefore, SP based studies focus on different various measures of reward and risk where the measure of reward is usually expected final wealth. Utility functions are used to combine the risk and reward into a single objective. In our SP model, we build the objective function as the maximization of the expected final wealth. We control the risk exposure by limiting the conditional Value-at-Risk (CVaR) within linear constraints. Changing the limiting parameters in these constraints would yield investment strategies for investors having different risk averseness. Different than the studies in literature, we employ risk control dynamically at each node of the scenario tree.

The model that can cover two other measures. The first one is the severity of the worst case scenario, which could be improved by a maximin approach (i.e., maximizing the wealth for the worst case scenario). The second one is to decrease the fluctuation of portfolio returns over the time-span of the scenario tree. This corresponds to the minimization of expected deviations among consecutive periods.

In summary, we aim to build a model which will be used to rebalance the current portfolio on hand such that the expected final wealth maximized over the planning horizon subject to the control on risk exposure (NOTE: The term rebalancing is usually used in conjunction with the fixed-mix rule to refer the trading process for restoring the constant mix. In this report, it refers to executing the buy/sell decisions suggested by the SP model).

The following is a list of the data categories that can be input to the portfolio optimization framework.

* Asset Prices/Returns: It is assumed that the current portfolio is a collection of a finite number of financial assets (e.g., stocks, bonds, cash, etc.). It should be noted that the term ‘asset’ may refer to ‘asset class’ since it is common in practice for some investors to invest in asset classes instead of individual assets. The data in this category are of two types:

1. Historical data: This set of data includes the historical return data for the universal set of assets and interest rates.
2. Future scenarios: This set of data is generated by the scenario generation algorithm. This is the most critical data set since it determines how well the model captures the uncertainty associated with the future asset returns.

* Cash inflows/outflows: Some assets generate positive cash flows to the investor (e.g., dividend payments of stocks) whereas some investors may have prescheduled or random cash outflows (e.g., liability flows) throughout the planning horizon.
* External and Internal Parameters: This group includes parameters those are external (e.g., transaction costs, interest rates) and those set by the investor (e.g., upper and lower bounds on individual asset positions, confidence level and limits for CVaR, objective weights, etc.).

2.1. A Brief Introduction to Scenarios

The asset returns (i.e., the realization component of the scenarios) are generated by sampling the historical data.

The generated asset returns are configured on a scenario tree, where we assume that:

* It is a symmetric tree, where the number of branches emanating from each node is the same for all nodes at the same stage.
* The number of branches emanating from each node decreases for the latter periods.

2.3. Probability Assignment through Moment Matching and Heteroskedasticity

Hoyland and Wallace (2001) use moment matching to assign probabilities to scenarios such that some statistical features of the generated scenarios comply with the statistical features of the historical data. In this study, the nonlinear optimization model given by (13) and (14) is solved taking the asset return scenarios and the corresponding probabilities as decision variables.

In this model, x is the vector of returns whereas p is the vector of scenario probabilities. The set of statistical features is represented by ST and fi(x, p) and SVi represents the i th statistical feature obtained from generated data and the historical data, respectively.

Our approach is based on the same setup; however, we incorporate a volatility clustering approach to model the state dependency of variances and covariances via Exponential Generalized Autoregressive Conditional Heteroskedasticity (EGARCH). In addition, we present an additional set of constraints to improve performance of the SP model to be run at the final step. Third, we consider a linear version of the optimization model for moment matching. Another difference is that we generate asset returns first and take only the probability values as decision variables in optimization instead of optimizing both return and probability simultaneously because the latter leads to a nonlinear model.

2.3.2. EGARCH

Having the asymmetric volatility as a major weakness of GARCH, another issue for most GARCH models is the non-negativity constraints on αi (i=0...q) and βi (i=1...p) for the purpose of obtaining positive 2 σ t for all t in all cases. These constraints make the estimation of GARCH models more difficult and create issues regarding the persistence of volatility over time. To overcome these issue Nelson (1991) introduced the EGARCH. A frequently used version of EGARCH is:

In contrast with most of the other ARCH models, the restrictions on the parameters (ω, α, β, and γ) are removed since their values have no effect on the sign of . The utilization of the logarithms ensures always having a positive conditional variance. A careful analysis of (22) reveals that is now a function of both the sign and the magnitude of , which enables the model to capture asymmetric volatility. Through the last term in the expression, EGARCH captures the magnitude effect of the shocks such that high (low) volatility periods are followed by high (low) volatility periods (i.e., volatility clustering).

The reason to select EGARCH to incorporate in our scenario generation algorithm is its well-known capability to capture the heteroskedastic behavior of financial time series data and cope well with asymmetric volatility. Relevant literature agrees on the superiority of the asymmetric GARCH models over regular GARCH models and even though there is not a single and 100% agreed-upon model, EGARCH turns out to be one of the most popular.

2.3.3. A New Set of Constraints for Probabilities

We introduce a new set of constraints for the usual moment matching model. The motivation behind this approach is to partially remove a potential shortcoming of the usual moment matching method that can be realized when multi-period models are involved.

Suppose that we generate a symmetric scenario tree for the SP model and let denote the number of single-period scenarios (i.e., nodes) emanating from a source node at time t-1 where t=1..T and T is the length of the horizon. We consider matching the moments for every single scenario set.

Every moment matching process assigns probabilities to the corresponding set of nodes, some of which can be zero because of the optimization. A low probability assigned in one of the preceding nodes leads to a low probability for the three-period scenario. In fact, if a node is assigned a probability of zero, then the rest of the tree emanating from that node is ignored by the SP model since the following three-period scenarios will assume a zero probability. For example, suppose that one of the four one-period scenarios for the first period of the tree in Figure 2 is assigned a zero probability; then 6 three-period scenarios emanating from that node will be disregarded by the SP model. In general, scenarios will be disregarded at the horizon when a one-period scenario assumes a zero probability at time t.

Even though zero probability scenarios are statistically acceptable to get close to the specified moments, this would produce an unwanted probability distribution at the horizon, where the expected wealth is computed. We believe that as the number of scenarios with zero (or close to zero) probabilities increase; the SP model will input fewer T-period scenarios, which might worsen the robustness of decisions led by the SP model. Therefore, we propose setting lower bounds for the probabilities during the moment matching process. The decision on the levels of bounds is an open question and higher bounds will obviously lead to large deviations from the target moments; however, these parameters can be computationally calibrated by back-testing given a set of data.

2.3.4. Algorithm Description: Alg-2

Suppose that ND represents the set of all nodes in the symmetric scenario tree excluding the leaf nodes. Suppose also that is the set of nodes that emanate from a particular node n. Once the scenario realizations are generated and the target moments are set, the problem is to assign probabilities for all the scenarios except the source node of the tree.

At this point there are two possible approaches to match the moments, first being to solve a large-scale optimization model that will assign probabilities to all scenarios in a single run. We prefer the alternative approach, which is based on solving one optimization model for each of the nodes in ND so that every set of scenarios is individually considered. Recalling the notation described in previous sections, the number of optimization models to be solved can be simply computed as

Using the previous notation, i is used as the index for risky assets, where we have m risky assets. The algorithm can be stated as follow:

1. **Generate asset return realizations for all nodes in the tree by randomly sampling the historical arithmetic returns (i.e., H)**
2. Estimate the third and fourth central moments, and (i=1..m), for all risky assets. Estimate the correlation matrix R among risky assets.
3. Estimate the univariate EGARCH(1,1) models for each risky asset to obtain parameters a,ω,α,β, and γ by maximum log-likelihood method. Where a SHOULD be the expected return of the stock from return formula:
4. Using the constant-mean model and the scenario realizations, obtain the error terms ε for each asset and node n∈ND
5. Using the estimated EGARCH(1,1) model and the error terms, predict for all risky assets the conditional variances for each n∈ND
6. Using the correlation matrix R and the conditional variances, compute the conditional covariance matrix for each n∈ND (See the explanation below for the constant correlation model).
7. Solve the optimization problem (23)-(30) to find the optimum probabilities for each n∈ND such that the weighted sum of distances to the target moments are minimized.

Min:

St:

In (23)-(30), and are the coefficients that capture the relative importance of the moments. The variables denoted by and are employed to compute the deviations.

The second moments are time-dependent and are computed conditionally on the previous periods’ realizations (i.e., scenarios) using EGARCH at every node throughout the scenario tree.

The correlation between assets has been addressed by researchers through Multivariate GARCH (MGARCH) models (for a recent and comprehensive review on MGARCH type modeling, see Bauwens et. al. (2006)). Even though these models are successful in capturing the conditional correlation among risky assets, an important shortcoming is the high number of parameters to be estimated, which brings in further computational concerns. Therefore, researchers proposed various versions of MGARCH models that provide less flexibility but come with simplicity. One of these studies is the **CCC-GARCH**. This approach decreases the number of parameters to be estimated significantly by **assuming that the correlations among assets are constant over time**. In CCC-GARCH, the conditional variances are computed via EGARCH and then the covariance matrix is constructed using the initial correlations and conditional variances.

Let be the mxm diagonal matrix formed by values where i=1..m. According to the CCC-GARCH model, the conditional covariance matrix for node n, is computed as . We use the elements of this covariance matrix in (28) to match covariance.

Constraint (29) puts a lower bound on the probability values for each node. The value of the parameter is subject empirical tests and can be computationally calibrated over a series of back-testing procedures. Even though we cannot claim a specific value, it is immediately realized that for all n. Therefore can be written as so that can be calibrated by experimenting with values of ∈ (0,1).

2.4. Stochastic Programming Model

**The outcomes that are of interest are the first stage variables representing the rebalancing decisions** (i.e., how much to buy/sell each asset at the time of running the model). Assuming that the investor continues investing over the discrete time periods, the process of managing the portfolio is based on updating the data sets and the initial portfolio and re-running the model at the beginning of each period.

2.4.1. Parameters

The main parameters used in the SP model can be grouped as:

* Asset return scenarios and scenario probabilities: This one period rates and probabilities are obtained via the scenario generation algorithm. They are the most critical parameter set since the model captures the real life uncertainty through these parameters.
* Economic parameters: Parameters such as transaction costs, interest rates and initial security prices can be considered in this group depending on the implemented investment scheme.
* User-defined parameters: Some parameters are set by the investor according to preferences or restrictions. Experimenting with these parameters may give valuable insight about how the model will behave in different environments (e.g., parameters controlling the risk averseness and exposure, statistical confidence levels, weights for different objectives, upper/lower bounds on asset positions, etc.).
* Cash flows: This group may contain dividend payments and possible liability obligations depending on the investment scheme.

NOTE: For simplicity of the presentation, the variables that apply to all risky assets are given as vectors shown in bold. Index i to will be appended later to represent the individual elements of the vector.

Sets:

s: Scenarios

t: Time periods

I = Set of monetary assets available (|I| = m)

: Set of scenarios for t = T (T is the horizon length)

Parameters:

y: Risk-free asset (cash) available at t = 0 (in monetary units)

**w: Vector for the amount of assets in the current portfolio (in monetary units)**

**Pts: Vector of security prices**

rbts: One period risk-free investment rate (if 1 period is 1 month, use 1-month rf rate)

**rts: Vector of one period returns for securities (if is 1 month, use 1 month returns)**

Lts: Liabilities or cash outflows

**cts: Vector of cash flows generated by securities (per unit share)**

ps: Probabilities of a scenario at t = T

ε: Unit transaction cost

**ub, lb: Vectors of upper and lower bounds for securities (weights in the portfolio)**

δ: Spread between borrowing and lending rates

βk: Weight of the k th objective in the objective function (EW, INS…)

α: Confidence level for CVaR.

LCVaR: Limit for CVaR

2.4.2. Variables

The variables in this model are of two types:

* Variables for decisions: In this set are the variables regarding the portfolio rebalancing and cash flow decisions. In other words, the decisions of how much to sell/buy each of the securities; how much to invest in risk-free asset; and how much to borrow/lend are made through these variables.
* Other variables: These variables are implied by the investment decisions such as expected wealth, auxiliary variables, etc

Since the optimization will be achieved within an SP model, we group the variables as first stage and stage variables.

2.4.2.1. First stage variables

These are the variables that will be practiced by the decision maker. They are time and scenario independent variables giving information on the optimum rebalancing decisions at t = 0 after the optimization model is run. The multi-period portfolio management framework is built on the successive running of the model at the beginning of each period and practicing the first stage variables immediately. These first stage variables can be regarded to have a time index as t = 0. The variables used in our model are:

**xb0: Vector of amounts of securities bought (in monetary units)**

**xs0: Vector of amounts of securities sold (in monetary units)**

**z0: Vector of amounts of securities held after purchase and selling (in monetary units)**

y0: Amount of cash invested in risk-free asset

b0: Amount of cash borrowed

2.4.2.2. Variables for stage t

Different than the first stage variables, variables for stage t are recourse variables which determine the corrective actions as future information is realized over the scenario tree after the first stage decisions are made. They are not actually practiced by the decision maker. The t th stage variables used in our model are:

**: Vector of amounts of securities bought (in monetary units)**

**: Vector of amounts of securities sold (in monetary units)**

**: Vector of amounts of securities held after purchase and selling (in monetary units)**

yts: Amount of cash invested in risk-free asset

bts: Amount of cash borrowed

2.4.2.3. Other Variables

These variables are used to get aggregate information on the outcomes of all investment decisions. This information might be used to achieve different types of performance analysis.

Ws: Wealth at the end of the horizon

Wt: Wealth at intermediate periods

: Expected positive deviation

: Expected negative deviation

EW: Expected final wealth

MWn: The variable used to limit the worst case scenario wealth.

INS: The variable used to measure the instability.

: Auxiliary variables used for CVaR constraints

2.4.3.1. Initialization and Restrictions

first stage variables are time and scenario independent. These first stage parameters must be linked to time dependent variables at t = 0. In other words, the time dependent variables are initialized to their first stage counterparts. Variables for sales/purchases and security holdings are initialized in (31)-(33). The cashrelated initializations are done through (34) and (35).

A very typical set of restrictions that might be posed by investors are the lower and upper bounds for the weights of particular assets in their portfolios. These might be personal, institutional or legal restrictions. Constraints (36) and (37) assure that the weights are within the allowed limits.

Another restriction is to prevent last time borrowing. In addition, no trading must take place at the very end of the planning horizon. Considering the planning horizon of T, any transaction will have to be closed at the end in order to evaluate the true performance of the portfolio. These restrictions can be imposed by the constraints in (38). Short-selling prevention and other sign restrictions are satisfied by (39).

2.4.3.2. Asset/Cash Balance

Variables in consecutive periods are connected through asset balance equations. Eq. (40) captures the relation between the current portfolio and the rebalanced portfolio using the first stage variables related to sell/purchase decisions. Eq. (41) captures the same relation between other time periods and scenarios.

Cash balance equations are given by (42) and (43). Eq. (42) builds the cash flow balance at t = 0, whereas (43) stands for the latter periods. In (43), the first term on the left is the investment in the risk-free asset (cash) and the second is the proceeds from dividend payments. The cash obtained from security sales are in the third term with applied transaction cost. The last term on the left is the borrowed cash, if any. The summation of these four terms is equal to the sum of outflows for cash investment, security purchase, liability obligation, and the pay-back for the previous periods borrowing, respectively. In the last equation (44) in this group, we compute the final wealth for each scenario for use in portfolio performance analysis.

2.4.3.3. Non-Anticipativity

Non-anticipativity constraints are given in (45)-(49). These constraints make sure that exploitation of the future information is prevented. **Construction of these constraints strictly depends on the scenario tree topology and needs specific attention.**

2.4.3.4. CVaR

We can briefly state our approach as the measurement and minimization of the downside risk of the portfolio.

Minimizing CVaR leads to solutions with low VaR and in fact they both result in the same decisions when the return-loss distribution is normal (Uryasev (2000)). It is the weighted average of VaR and losses exceeding VaR. Therefore, CVaR ≥VaR.strictly holds

**Apart from the notation presented previously in this text**, let f(x,y) be the loss function associated with decision vector x and random vector y. Supposing that y has a density, the cumulative function of this loss function is:

Given the confidence level α,

Define

Then is obtained as

If y has discrete scenarios with probabilitiesπ j instead of a continuous distribution, then (53) can be approximated as

Then, using auxiliary variables uj, CVaR is restricted above by w through the constraints (56) and (57).

Suppose that an investor defines his loss function as the difference between the initial portfolio value and the final wealth, which is obtained through discrete scenarios and denoted by Ws. Therefore, given the confidence level of α and upper limit LCVaR, constraints (58)-(60) impose control CVaR.

It should be noted that the actually realized CVaR may not be strictly controlled by LCVaR since, constraints (58)-(60), as all other constraints in the model, assume that the inputted scenario tree reflects the true future uncertainty (i.e., the true scenario tree). Since the generated scenario tree is an approximation, constraints (58)-(60) provide an approximate risk control.

2.4.3.5. Worst Case Scenario

Investors benefit from robustness especially when out-of-sample scenarios come true. It should be noted that not all out-of-sample scenarios are critical in terms of robustness since some of them may be captured to some extent by the in-sample scenarios. Plus, robustness cannot remedy all problems caused by an out-of-sample scenario but can decrease the realized severity according to the structure of the scenario.

SP models can not include all possible scenarios due to complexity issues but approximate the actual universal set of scenarios in order to achieve optimization under uncertainty. Therefore, depending on the scenario generation process, there might be a tradeoff between the robustness and computational complexity. Given a fixed number of scenarios, an SP model based approach may increase the robustness by including some worst-case scenarios and optimizing the objective function for those unexpected scenarios. The worst case scenario results in the minimum objective value (for a maximization problem); therefore, maximizing the minimum possible outcome may improve the robustness of the decisions.

2.4.3.6. Dynamic Risk Control

Measuring and controlling the risk of an investing strategy merely at the end of the horizon may lead to unwanted results. If risk control is achieved only to the end of the horizon and a multi-period model with a scenario tree is utilized, the model would result in decisions that may lead to deviations from the targeted risk exposure in the former periods as long as performance constraints at the end of the horizon are not violated. Therefore, the performance of the resulting trading will be highly dependent on the first stage decisions. If risk is not controlled within the first period, then the first stage variables will assume values such that the objective at the end will be optimized, which may lead to high-risk exposure for the first period. Repetition of the investment process eventually creates high risk exposure in the long run even if it is controlled in the last period of the model.

To decrease the deviation from the desired risk level, the proposed model constrains CVaR for each node of the scenario tree, other than the leaf nodes. In other words, risk is controlled by limiting CVaR at each decision epoch considering the one period ahead risk exposure and, in turn, spreading risk control over the whole scenario tree.

Can be illustrated via a simple case. Suppose that is a decision vector that corresponds to node n and ρ is a risk measure such that ρ () is the risk exposure caused by decision vector . Let RLn be the risk limit to be imposed on the decisions for node n. Then the constraints given by (61) provide a more powerful risk control over the scenario tree than the case where risk control is accomplished merely at the horizon over the leaf nodes.

Recalling ND represents the set of all nodes in the scenario tree for the proposed SP model excluding the leaf nodes and that is the set of nodes that emanate from node n, the idea of dynamic risk control can be combined with the linear CVaR constraints as in (62)-(64) to be incorporated into the SP model. Note that we will use index n instead of s for simplicity.

In (62)-(64), t represents the time period that node n implies in the tree and is the probability of the scenario represented by node j, which emanates from node n such that . LCVAR is the same for all nodes; however, different values can be used to obtain different risk limits for different periods and/or different nodes, which might bring in further flexibilities in risk control. We build the loss function for CVaR to be the one-period-ahead loss given by:

The same setting can be used for other risk measures such as maximizing the wealth for the worst possible scenario after each node. For each node n, denote by MWn the minimum wealth resulting from the decision vector of node n, which can be placed in the objective for maximization and constrained above by the wealth for each scenario emanating from node n as in (65).

2.4.3.7. Stability over the Horizon

Another performance measure that might be of interested is the fluctuation of the portfolio value over time. Given a fixed horizon, less fluctuating portfolio returns over oscillating portfolio returns if the final wealth levels do not differ significantly might be preferable. It should be noted that this measure obviously has overlapping characteristics with other risk measures such as variance or volatility. However, direct minimization of these measures implies nonlinearity. The stability can be improved by minimizing the expected deviations between the portfolio returns of consecutive periods within a linear approach as in (66)-(69) where INS denotes an average measure for instability.

Where **should be** the standard deviation values in + and – about the distribution of probability for the wealth at time, multiplied by the said wealth (or the previous wealth…). Note that being expected values, , which means that

2.4.3.8. Objective

Like many other portfolio management models, maximizing the expected return of the portfolio at the end of the horizon is the major objective. To be able to consider the expected return in the objective function, (70) is included in the constraint set.

Following the discussions in Sections 2.4.3.5 and 2.4.3.7, a general objective function can be written as:

where β1, β2, and β3 are the weights preferred by the investor that measure the relative importance of expected final wealth, worst case scenario improvement, and stability, respectively. Therefore, a comprehensive SP model can be given by:

Max: (71)

Subject to (31)-(39) and (62)-(70)

MULTI STAGE MODELING

At period t = 1 a certain amount of wealth is available in assets 1 = 1, …, n and cash which we index as n + 1.

We denote to be the currency value of the initially available assets.

The decision maker must decide each period how to rearrange his portfolio to achieve best return on his initial investment over time. We define time steps t = 1, …, T by months. At each time period t, the investor can either hold on asset I, buy more or sell off part (or all) of asset i.

We denote the amount sold of asset i in period t and by the amount held on to. Selling means decreasing the value of and increasing the value of cash, . Also, the investor can decide to use his cash to buy certain amounts of asset i. The amount bought is denoted with .

Buying and selling causes transaction costs, which we assume to be proportional (we need them to be fixed…) to the amount of currency value of asset traded. We denote by the transaction costs associated with buying one unit of I and with the costs of selling. So, buying requires units of cash and selling provides units of cash.

Once rebalancing is done, new holdings can be calculated.

Each period, returns on assets for next period t+1, , are not known with certainty, only return on cash, , is known. However, we assume we know the probability distribution for . While the decision at t has to be made based on said distributions, the values of prior returns, , have already been observed.

We denote the n-dimensional random vector with outcomes , with the corresponding probability and the set of all possible outcomes in t.

After the last period T, no decision is made. Only the value of the portfolio is determined. We call this value . The goal is, however, to maximize , the expected utility of the value of the portfolio after period T. the utility function describes the way the investor views risk. Nonlinear utility functions require nonlinear programming techniques. Our methodology is not restricted to linear problems. However, we approximate the nonlinear function by a piecewise linear function with sufficiently large number of linear segments.

In the model, short selling is not considered, although it could be easily implemented. We also do not consider borrowing of cash, which can also be implemented easily.

Upper and lower bound on holdings, as well as on amount of assets to be sold or bought are placed: ub, lb, sub, slb, bub, blb; given by investor, or market. In general, we formulate:

Where

….

PYTHON TREE BUILDING LIBRARY

2 Nested Formulation

In multistage setting, the underlying data process is revealed sequentially. The data process is modeled as a stochastic process so that ξ1 is deterministic and ξ2, . . ., ξT is to be revealed over time.

Decision variables in each stage are categorized into state variables and control variables, denoted by x and y respectively. The decision variables x1, y1 are deterministic and the initial value of state variable x0 is supposed to be known.

16.2 Multi-period portfolio optimization

A portfolio manager oversees multiple assets (i = 1, . . . , N) and a bank account (i = N+1). For a specified number T of stages, the manager wants to maximize his utility by dynamically rebalancing the portfolio. Let {rit} be the return process of asset i. At the end of each period, the position of i th asset xit equals the start position xi,t−1, plus the realized return ritxi,t−1 during the period, plus the newly long positions bit, minus the newly short positions sit. Transaction costs are fb, fs for buying and selling respectively. The capital in the bank account will be adjusted accordingly. We consider a simple asset pricing model that decomposes the excess return as the return explained by Capital Asset Pricing Model (CAPM), alpha and idiosyncratic risk